Programmable Controller

PLC Board

Controlador Programable

Tarjeta PLC

Controlador Programável

Cartão PLC

PLC-01

User´s guide

Guia del usuario

Manual do usuário

MANUAL
DO CARTÃO PLC1.01

PROGRAMÁVEL EM LINGUAGEM LADDER PELO SOFTWARE WLP

08/2004

Software da PLC1: V1.5X

Software WLP: V4.0X

0899.4921 P/3

A informação abaixo descreve as revisões ocorridas neste manual.

Revisão	Descrição da revisão	Capítulo
1	Primeira revisão	-
2	Funções Relativas a Monitoração On-line	11

Índice

1 Referência Rápida dos Parâmetros 04
2 Mensagens de Erro 06

CAPÍTULO 1
Introdução

CAPÍTULO 2
Instruções de Segurança

CAPÍTULO 3
Informações Gerais

CAPÍTULO 4
Características Gerais da PLC1

CAPÍTULO 5
Descrição dos Conectores

CAPÍTULO 6
Configuração do conversor CFW-09 com a placa PLC

CAPÍTULO 7
Instalação da placa no conversor

CAPÍTULO 8
Descrição Detalhada dos Parâmetros

CAPÍTULO 9
WLP

CAPÍTULO 10
Blocos do WLP

CAPÍTULO 11
Monitoração On-Line Utilizando WLP

CAPÍTULO 12
Protocolo Modbus na PLC1

REFERÊNCIA RÁPIDA DOS PARÂMETROS, MENSAGENS DE ERRO

Software: V1.5X
Aplicação:
Modelo:
N.º de série:
Responsável:
Data: / / .

A faixa de parâmetros que vai de 750 a 899, totalizando 150. Os 50 primeiros, são pré-definidos pelo sistema ou reservados. Os 100 restantes são de uso geral, ou seja, podem ser programados pelo usuário.

Segue abaixo a descrição dos parâmetros do sistema que estão definidos.

1. Parâmetros

Parâmetro	Descrição	Faixa de Valores	Ajuste de Fábrica	Unidade	Página
P750	Versão do firmware da PLC1	Leitura			
P751	Ciclo de scan em unidades de 100µs	Leitura		x100µs	
P752 (*)	Zera marcadores retentivos	0...1	0		
P753 (*)	Carrega valores de fábrica, se =1234	0..65535	0		
P754	Referência de posição (rotações)	Leitura		rot	
P755	Referência de posição (fração de volta)	Leitura		graus / 10	
P756	Sinal da posição real 0 = negativo 1 = positivo	Leitura			
P757	Posição real (rotações)	Leitura		rot	
P758	Posição real (fração de volta)	Leitura		graus / 10	
P760	Kp: ganho proporcional de posição	0...200	50		
P761	Ki: ganho integral de posição	0...200	0		
P762	Erro de lag máximo	0...65535	0	graus / 10	
P763	Desabilita programa do usuário se =1	0...1	0		
P764 (*)	Endereço da PLC na rede	1...247	1		
P765 (*)	Baud rate da RS232	1 = 1200bps 2 = 2400bps 3 = 4800bps 4 = 9600bps 5 = 19200bps	4	bits / segundo	
P766	Estado das entradas digitais	Leitura			
P767 (*)	Velocidade síncrona do motor em RPM	0...10000	1800	RPM	
P768 (*)	Número de pulsos do encoder	0...65535	1024	ppr	
P769 (*)	Posição do pulso de nulo do encoder	0...3599	0	graus / 10	

(*) IMPORTANTE: o sistema precisa ser reinicializado quando um ou mais desses parâmetros for alterado, para que atue conforme o programado.

Parâmetro	Descrição	Faixa de Valores	Ajuste de Fábrica	Unidade	Página
P772	Baudrate da CAN	0=1Mbit/s 1=Reservado 2=500 Kbit/s 3=250 Kbit/s 4=125 Kbit/s 5=100 Kbit/s 6=50 Kbit/s 7=20 Kbit/s 8=10 Kbit/s	0	bits/segundo	
P773	Recuperar bus off	0=Manual 1=Automático			
P775	Status da CAN	Leitura			
P776	Contador de telegramas recebidos	Leitura			
P777	Contador de telegramas transmitidos	Leitura			
P778	Contador de erros detectados	Leitura			

2. Mensagens de Erro

Indicação	Significado	Observação
E50	Erro de lag	Erro fatal, desabilita o inversor. Ver parâmetro P762.
E51	Falha ao gravar programa	Reinicializar o sistema e tentar novamente.
E52	Dois ou mais movimentos habilitados simultaneamente	Verificar a lógica do programa do usuário.
E53	Dados de movimento inválidos	Provavelmente algum valor zerado de velocidade, aceleração, etc.
E54	Inversor desabilitado	Tentativa de executar um movimento com o inversor desabilitado
E55	Programa incompatível ou fora dos limites da memória	Verificar programa e reenviá-lo. Esse erro também ocorre quando não há programa na PLC (primeira vez que a mesma é energizada).
E56	CRC errado	Transmitir novamente.
E57	Eixo não referenciado para movimentação absoluta	Antes de um movimento absoluto, uma busca de zero de máquina deve ser executada.
E58	Falta de referência do mestre	Erro fatal: após estabelecida comunicação inicial, entre mestre e escravo, por algum motivo a mesma foi interrompida.
E61	Bus off	*Bus off* detectado no barramento CAN, devido a um grande número de erros de transmissão, seja por problemas no barramento ou instalação inadequada.

Obs: nos erros fatais, E50 e E58, o inversor é desabilitado e precisa ser reinicializado.

INTRODUÇÃO

O cartão PLC1 agrega ao inversor CFW-09, funções importantes de CLP (Controlador Lógico Programável), possibilitando a execução de complexos programas de intertravamento, que utilizam as entradas e saídas digitais do cartão, bem como as entradas e saídas digitais e analógicas do próprio inversor, que podem ser acessadas pelo programa do usuário.

Dentre as várias funções disponíveis, podemos destacar desde simples contatos e bobinas até funções utilizando ponto flutuante, como soma, subtração, multiplicação, divisão, funções trigonométricas, raiz quadrada, etc.

Outras funções importantes são blocos PID, filtros passa-alta e passa-baixa, saturação, comparação, todos em ponto flutuante.

Além das funções citadas acima, a PLC1 oferece blocos para controle de posição e velocidade do motor, que são posicionamentos com perfil trapezoidal, posicionamentos com perfil S, geração de referência de velocidade com rampa de aceleração trapezoidal, etc. (obs.: para posicionamento, é imperativo o uso de um encoder acoplado ao motor).

Todas as funções podem interagir com o usuário, através dos 100 parâmetros programáveis, que podem ser acessados diretamente pela IHM do inversor e, através do WLP, podem ser customizados com textos e unidades do usuário.

Através das novas funções modbus introduzidas a partir da versão V1.50 da placa PLC1, é possível executar funções avançadas de monitoração on-line no software WLP a partir da versão V4.00.

ATENÇÃO!
A versão de software do inversor **CFW-09** deve ser a **V2.40** ou superior.

INSTRUÇÕES DE SEGURANÇA

Este manual contém as informações necessárias para o uso correto da placa PLC1 com o inversor de freqüência CFW-09.

Ele foi escrito para ser utilizado por pessoas com treinamento ou qualificação técnica adequados para operar este tipo de equipamento.

Somente pessoas com qualificação adequada e familiaridade com o inversor CFW-09 e equipamentos associados devem planejar ou implementar a instalação, partida, operação e manutenção deste equipamento.

Estas pessoas devem seguir todas as instruções de segurança contidas neste manual e/ou definidas por normas locais.

Não seguir as instruções de segurança pode resultar em risco de vida e/ou danos no equipamento.

NOTA!
Para os propósitos deste manual, pessoas qualificadas são aquelas treinadas
de forma a estarem aptas para:

1. Instalar, aterrar, energizar e operar o CFW-09, bem como a placa PLC1, de acordo com este manual e os procedimentos legais de segurança vigentes;
2. Usar os equipamentos de proteção de acordo com as normas estabelecidas;
3. Prestar serviços de primeiros socorros.

PERIGO!
Sempre desconecte a alimentação geral antes de tocar qualquer componente
elétrico associado ao inversor.
Altas tensões e partes girantes (ventiladores) podem estar presentes mesmo após a desconexão da alimentação. Aguarde pelo menos 10 minutos para a descarga completa dos capacitores da potência e parada dos ventiladores.
Sempre conecte a carcaça do equipamento ao terra de proteção (PE) no ponto adequado para isto.

ATENÇÃO!
Os cartões eletrônicos possuem componentes sensíveis a descargas eletrostáticas. Não toque diretamente sobre componentes ou conectores. Caso necessário, toque antes na carcaça metálica aterrada ou utilize pulseira de aterramento adequada.

NOTA!
Leia completamente este manual antes de instalar ou operar o cartão com o inversor.

INFORMAÇÕES GERAIS

Este capítulo fornece informações sobre o conteúdo deste manual e o seu propósito.

3.1 SOBRE O MANUAL

Este manual descreve basicamente os procedimentos necessários para a instalação do WLP, a criação de projetos e fornece uma visão global dos blocos existentes na PLC1.
Cap. 1- Introdução;
Cap. 2- Instruções de Segurança;
Cap. 3- Informações Gerais;
Cap. 4- Características Gerais da PLC1;
Cap. 5- Descrição dos Conectores;
Cap. 6- Configuração do conversor CFW-09 com a placa PLC;
Cap. 7- Instalação da placa no conversor;
Cap. 8- Descrição Detalhada dos Parâmetros;
Cap. 9- WLP;
Cap. 10- Blocos do WLP;
Cap. 11- Monitoração On-Line Utilizando WLP;
Cap. 12- Protocolo ModBus na PLC1.

O propósito deste manual é dar as informações necessárias para o bom uso da PLC1. Devido à grande gama de funções deste produto, é possível aplicá-lo de formas diferentes das apresentadas aqui. Não é a intenção deste manual esgotar todas as possibilidades de aplicação do cartão, nem a WEG pode assumir qualquer responsabilidade pelo uso da PLC1 baseado neste manual.

É proibida a reprodução do conteúdo deste manual, no todo ou em partes, sem a permissão por escrito da WEG.

CARACTERÍSTICAS GERAIS DA PLC1

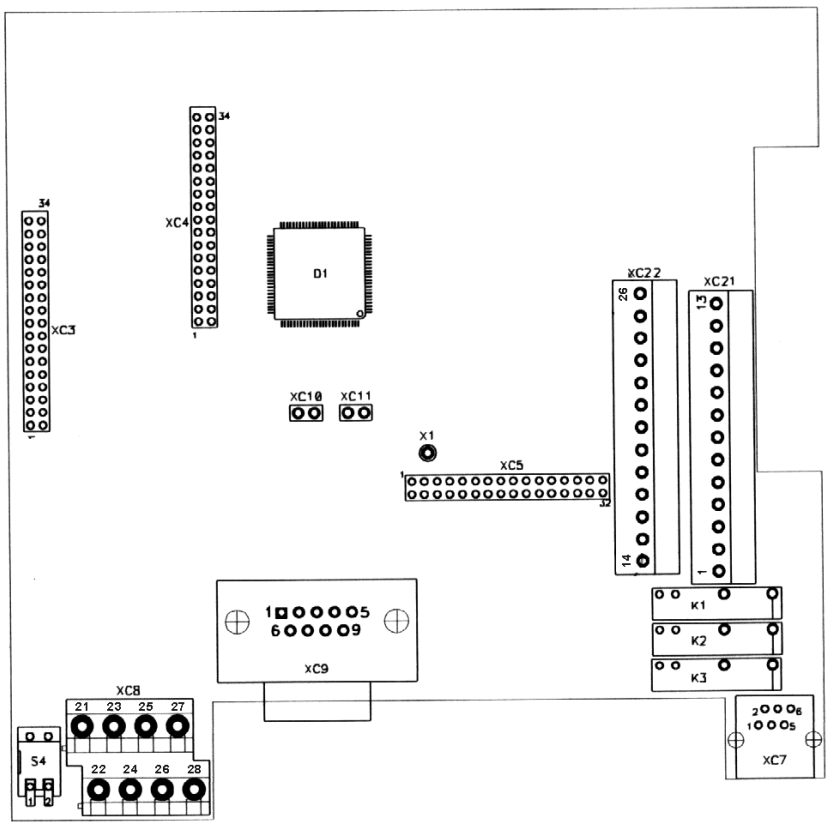
4.1 Hardware

- ☑ 9 entradas digitais isoladas, bidirecionais, 24VCC
- ☑ 3 saídas digitais a relé 250V x 3A
- ☑ 3 saídas digitais optoacopladas, bidirecionais, 24VCC x 500mA
- ☑ 1 entrada de encoder isolada, com alimentação externa entre 18 e 30VCC
- ☑ Alimentação para o encoder 15VCC x 300mA
- ☑ 1 interface de comunicação serial – RS-232C (Protocolo padrão: MODBUS-RTU)
- ☑ Compatível com todas as mecânicas do CFW-09
- ☑ Programação do usuário em linguagem Ladder, com blocos específicos para posicionamento e funções de CLP
- ☑ Permite o uso das entradas e saídas digitais e analógicas do CFW-09, o que totaliza 15 entradas digitais, 9 saídas digitais, 2 entradas analógicas e 2 saídas analógicas, acessadas pelo ladder;

4.2 Software

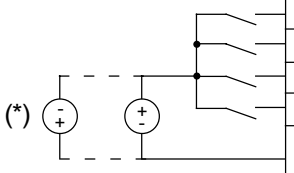
- ☑ Faixa de parâmetros que vai de 750 a 899, totalizando 150. Os 50 primeiros, são pré-definidos pelo sistema ou reservados. Os 100 restantes são de uso geral, ou seja, podem ser programados pelo usuário podendo ser usados para diversas funções, como contadores, timers, referência de velocidade, aceleração, posição, etc.
- ☑ Marcadores do tipo BIT e WORD voláteis (inicializados em zero) e retentivos, e do tipo FLOAT volátil.
- ☑ A programação da placa é feita através do programa WLP, utilizando ladder. Com o WLP V4.00 ou posterior e firmware PLC1 V1.50 ou posterior esse mesmo ladder pode ser monitorado on-line.
- ☑ Capacidde de memória para o programa do usuário: 64kB (65536 bytes)

DESCRIÇÃO DOS CONECTORES



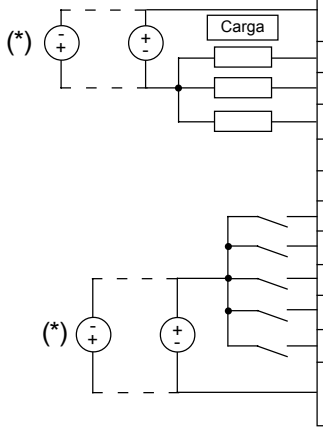
Conector XC21: Saídas a relé e entradas digitais

Conector XC21			Função	Especificações
1	C	DO1	Saídas digitais a relé	Capacidade dos contatos: 3A 250VAC
2	NA			
3	C	DO2		
4	NA			
5	C	DO3		
6	NA			
7	NC		Não conectado	
8	NC		Não conectado	
9	DI6		Entradas digitais isoladas	Tensão de entrada: 15...30VCC Corrente de entrada: 11mA @ 24VCC
10	DI7			
11	DI8			
12	DI9			
13	COM DI		Comum das entradas DI6...DI9	



ATENÇÃO!
(*) Fonte de alimentação externa

Conector XC22: Saídas a transistor e entradas digitais 24V



Conector XC22		Função	Especificações
14	NC	Não conectado	
15	COM DO	Comum das saídas digitais DO4, DO5 e DO6	
16	DO6	Saídas digitais opto-isoladas bidirecionais	Tensão máxima: 48VCC Capacidade de corrente: 500mA
17	DO5		
18	DO4		
19	NC	Não conectado	
20	NC	Não conectado	
21	DI1	Entradas digitais isoladas bidirecionais	Tensão de entrada: 15...30VCC Corrente de entrada: 11mA @ 24VCC
22	DI2		
23	DI3		
24	DI4		
25	DI5		
26	COM DI	Comum das entradas DI1...DI5	

ATENÇÃO!
(*) Fontes de alimentação externas

Conector XC3: Placa Profibus da HMS

Possibilita a PLC de comunicar-se através de rede Profibus.

Conector XC7: RS-232C

Conector XC7		Função	Especificações
1	5VCC	Alimentação de 5VCC	Capacidade de corrente: 50mA
2	RTS	Request to send	
3	GND	Referência	
4	RX	Recebe	
5	GND	Referência	
6	TX	Transmite	

Conector XC8: Entrada do 24VCC externo e rede CANopen

Conector XC8		Função	Especificações
21	CAN GND	GND da CAN	
22	24VCC	Alimentação para entrada de encoder	18..26Vcc Corrente consumida: 25mA + a corrente do encoder.
23	CANL	CANL	
24	GND ENC	Referência dos 24Vcc do encoder	
25	CANH	CANH	
26	NC	Não conectado	
27	CAN 24Vcc	Alimentação para rede CANopen	18..26Vcc 50mA @ 24Vcc
28	NC	Não conectado	

Conector XC9: Encoder Incremental

Nas aplicações que necessitam de maior precisão de velocidade ou aplicações de posicionamento, é necessária a realimentação da velocidade do eixo do motor através de encoder incremental. A conexão ao inversor é feita através do conector XC9 (DB9) do cartão PLC1.

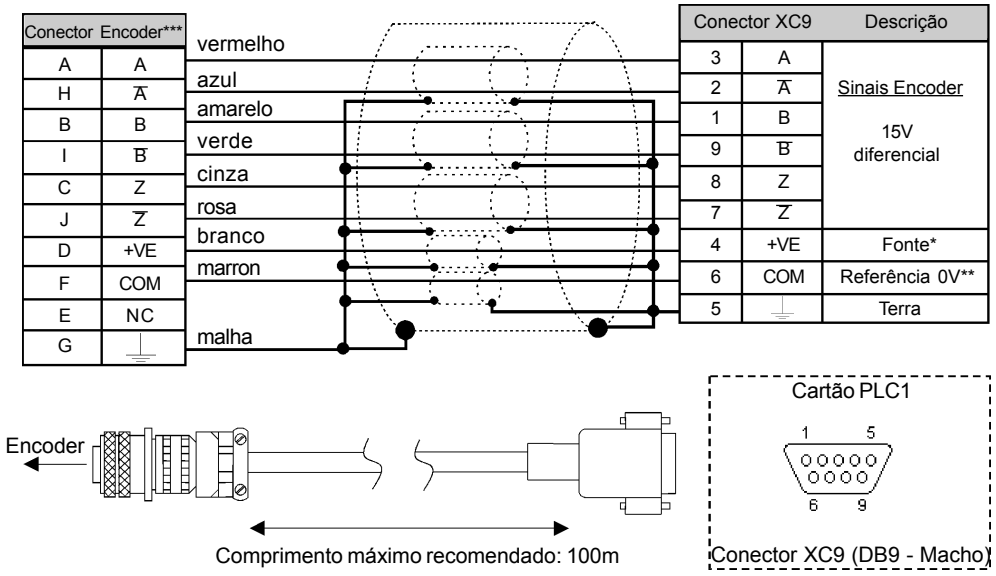
O encoder a ser utilizado deve possuir as seguintes características:
- ☑ Tensão de alimentação: 15 V, com consumo menor que 200 mA;
- ☑ 2 canais em quadratura (90º) + pulso de zero com saídas complementares (diferenciais): Sinais A, $\overline{A}$, B, $\overline{B}$, Z e $\overline{Z}$;
- ☑ Circuito de saída tipo "Linedriver" ou "Push-Pull" (nível 15V);
- ☑ Circuito eletrônico isolado da carcaça do encoder;
- ☑ Número de pulsos por rotação recomendado: 1024 ppr;

Na montagem do encoder ao motor seguir as seguintes recomendações:

- ☑ Acoplar o encoder diretamente ao eixo do motor (usando um acoplamento flexível, porém sem flexibilidade torsional);
- ☑ Tanto o eixo quanto a carcaça metálica do encoder devem estar eletricamente isolados do motor (espaçamento mínimo: 3 mm);
- ☑ Utilizar acoplamentos flexíveis de boa qualidade que evitem oscilações mecânicas ou "backlash";

Para a conexão elétrica utilizar cabo blindado, mantendo-o tão longe quanto possível (>25cm) das demais fiações (potência, controle, etc.). De preferência, dentro de um eletroduto metálico.

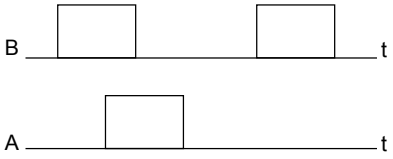
Durante a colocação em funcionamento é necessário programar o parâmetro **P202** - Tipo de controle = 4 (Vetorial c/ Encoder) para operar com realimentação de velocidade por encoder incremental.



* Fonte de alimentação 15V / 220mA para encoder
** Referenciada ao terra via 1μF em paralelo com 1kΩ
*** Pinagem válida p/ encoder HR526xxxxB5-Dynapar. Para outros modelos de encoder verificar a conexão correta para atender a sequência necessária.

NOTA!
A freqüência máxima do encoder permitida é 100kHz.

Seqüência necessária dos sinais do Encoder:



Motor girando no sentido horário

Jumper XC10: Gravação do firmware

Jumper XC10	
Aberto	Funcionamento normal
Fechado	Gravação de firmware

Jumper XC11: Erro de encoder

Jumper XC11	
Aberto	Habilita geração de erro de encoder
Fechado	Não gera erro de encoder

CONFIGURAÇÃO DO CONVERSOR CFW-09 COM A PLACA PLC

☑ Tipo de controle (P202):

Para os blocos que geram referência de velocidade (JOG e SETSPEED) pode-se usar o inversor no modo 'Sensorless' (**P202=3**), lembrando que nesse modo, não há muita precisão em baixas velocidades. Além disso, o ganho Kp, de posição (P760) deve ser zerado, para não causar instabilidade no momento que o motor é habilitado. Para os blocos de posição (TCURVE e SCURVE) o inversor deve operar no modo vetorial com encoder (**P202 = 4**).

Observações importantes:

- ☑ sempre que possível usar o modo vetorial com encoder;
- ☑ evitar os modos escalares (V/F) se a PLC vai gerar referência de velocidade;
- ☑ verificar o correto ajuste dos parâmetros P161 e P162 que são o ganho proporcional de velocidade e o ganho integral de velocidade, respectivamente, eles são fundamentais para um bom desempenho do inversor.

☑ Seleção Local / Remoto (P220):

Quando a PLC é usada para geração de movimento, esta opção deve ficar como 'Sempre local' (**P220=0**).

☑ Seleção Referência Local (P221):

Quando a PLC é usada para geração de movimento, esta opção deve ficar como 'PLC' (**P221=11**), ou seja, a referência de velocidade será dada pela placa PLC.

☑ Seleção Gira/Pára Local (P224):

Para que a PLC possa controlar o conversor, em relação a girar e parar e também habilitar e desabilitar o drive, essa opção deve ficar em 'PLC' (**P224=4**).

☑ Função Saída AO1 (P251):

Para que a saída analógica 1 (AO1) do inversor possa ser controlada pela PLC, setar P251=12. Observar P252 que é o ganho da saída analógica 1.

☑ Função Saída AO2 (P253):

Para que a saída analógica 2 (AO2) do inversor possa ser controlada pela PLC, setar P253=12. Observar P254 que é o ganho da saída analógica 2.

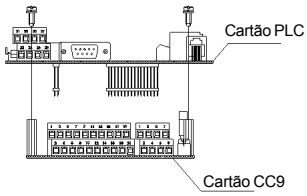
☑ Entradas digitais DI101...DI106, P263...P268:

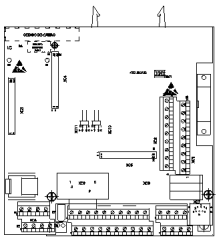
Correspondem às entradas digitais DI1...DI6 do inversor e são lidas pela PLC, independentemente da função programada nos parâmetros P263...P268.

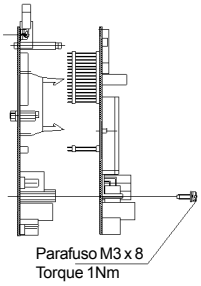
☑ Saídas digitais a relé DO101...DO103, P277, P279 e P280:

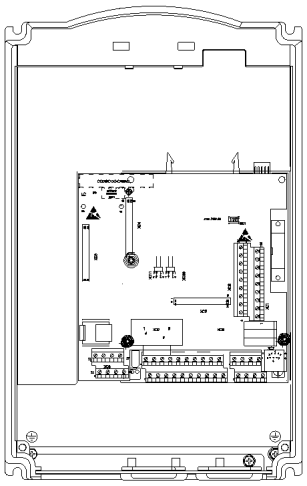
Correspondem às saídas RL1...RL3 do drive. Para que estas saídas sejam controladas pela PLC, é necessário que sejam programadas para função ‘PLC’, ou seja P277=27, P279=27 e P280=27.

INSTALAÇÃO DA PLACA NO CONVERSOR

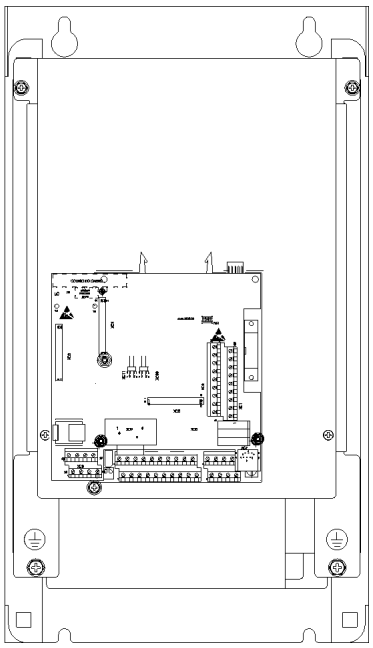








CFW-09 Mecânicas 1 e 2



CFW-09 Mecânicas 3 a 10

NOTA!

Se o conversor utilizado for da mecânica 1 (correntes de 6 a 13A em tensões de rede entre 220-230V ou correntes 3.6 a 9A em tensões de rede entre 380-480V), a lateral plástica do conversor deve ser removida para que a PLC possa ser encaixada corretamente.

Em qualquer outra mecânica, a PLC pode ser encaixada diretamente.

DESCRIÇÃO DETALHADA DOS PARÂMETROS

Parâmetro	Faixa [Ajuste fábrica] Unidade	Descrição / Observações
P750 Versão do firmware da placa PLC	- [-] -	☑ Parâmetro de leitura. Exemplo: versão 1.30, lê-se no parâmetro o valor 130.
P751 Ciclo de scan do programa do usuário	- [-] x100 µs	☑ Parâmetro de leitura. Mostra a duração do ciclo do programa do usuário, cada unidade corresponde a 100µs. Uma maneira fácil de obter-se o valor do ciclo de scan em milisegundos, é dividir o valor de P751 por 10. Exemplo: lido um valor de 79, significa que o ciclo de scan do programa é de 79 ÷ 10 = 7,9ms.
P752 (*) Zera marcadores retentivos	0...1 [0] -	☑ Zera marcadores retentivos, tanto do tipo bit, como do tipo word. Deve-se colocar 1 (um) no parâmetro e reinicializar o sistema. O valor deste parâmetro volta para 0 (zero) automaticamente.
P753 (*) Carrega valores de fábrica, se =1234	0...65535 [0] -	☑ Carrega valores de fábrica para os parâmetros de sistema (P750...P799). Colocar o valor de 1234 nesse parâmetro e resetar o sistema.
P754 Referência de posição (rotações)	- [-] rot	☑ Mostra posição de referência em rotações. A posição de referência começa em zero e após a conclusão do movimento, volta para zero.
P755 Referência de posição (fração de volta)	- [-] graus/10	☑ Mostra fração de volta da posição de referência em décimos de grau. A posição de referência começa em zero e após a conclusão do movimento, volta para zero.
P756 Sinal de posição	- [-] -	☑ Sinal da posição real, mostrada nos parâmetros P757 e P758. 1 = positivo e 0 = negativo.
P757 Posição real (rotações)	- [-] rot	☑ Mostra posição real em rotações.
P758 Posição real (fração de volta)	- [-] graus/10	☑ Mostra fração de volta da posição real em décimos de grau.
P760 Ganho proporcional (Kp) de posição	0...200 [50] -	☑ Aumentar esse ganho para deixar a resposta a um erro de posição, mais rápida, diminuí-lo caso o sistema vibre, ou torne-se instável.
P761 Ganho integral (Ki) de posição	0...200 [0] -	☑ Tem a função de zerar eventuais erros de posição. Normalmente, esse ganho é zero, pois pode causar overshoot de posição, ou seja, passar da posição desejada e retornar.

Parâmetro	Faixa [Ajuste fábrica] Unidade	Descrição / Observações
P762 Erro de lag máximo	0...65535 [0] graus/10	☑ É o erro de posição máximo permitido em posicionamentos, ou seja, a máxima diferença entre a posição de referência e a posição real, em graus. O valor do parâmetro é o lag dividido por 10. Por exemplo um valor de 10 em P762, significa que o máximo erro de seguimento é 1 grau. Se P762 = 0 (valor default) o erro de lag não será verificado.
P763 Desabilita o programa do usuário se=1	0...1 [0] -	☑ Desabilita o programa do usuário, se for programado em 1. Somente deve ser usado em alguma situação anormal, em que o programa esteja causando algum tipo de erro que, por exemplo, impeça a comunicação com a interface serial. Nesse caso, desabilita-se o programa, carrega-se a versão corrigida e então habilita-se novamente.
P764 (*) Endereço da PLC na rede	1...247 [1] -	☑ Em caso de ligação em rede MODBUS, através de uma RS485 (via conversores RS232-RS485), por exemplo, esse parâmetro define o endereço da placa na rede.
P765 (*) Baud rate da RS232	1...5 [4 (= 9600bps)] -	☑ Ajusta o baud-rate da interface serial. Os valores permitidos são: **P765** / **Baud-Rate (bps)** 1 / 1200 2 / 2400 3 / 4800 4 / 9600 5 / 19200
P766 Estado das Entradas Digitais	- [-] -	☑ Mostra o status das 15 entradas digitais, ou seja as 9 da PLC1 mais as 6 do conversor. O número lido deve ser convertido para binário, daí tem-se uma leitura direta do estado de cada entrada. BIT14 BIT13 BIT12 BIT11 BIT10 BIT9 BIT8 DI101 DI102 DI103 DI104 DI105 DI106 DI9 BIT7 BIT6 BIT5 BIT4 BIT3 BIT2 BIT1 BIT0 DI8 DI7 DI6 DI5 DI4 DI3 DI2 DI1 Onde as DI101 a DI106, representam o estado das 6 entradas digitais do drive e as DI1 a DI9, representam o estado das 9 entradas digitais da PLC1.
P767 (*) Velocidade síncrona do motor	0...10000 [1800] rpm	☑ Por exemplo, um motor de 4 pólos em 50Hz possui uma velocidade síncrona de 1500RPM.
P768 (*) Resolução do encoder	0...65535 [1024] ppr	☑ É o número de pulsos por rotação do encoder.
P769 (*) Posição do pulso nulo do encoder	0...3599 [0] graus/10	☑ O valor entrado deve ser em décimos de grau. Pode ser utilizado na busca de zero de máquina, a fim de adiantar a posição de zero.

(*) IMPORTANTE: o sistema precisa ser reinicializado quando um ou mais desses parâmetros for alterado, para que atue conforme o programado.

P772
Baudrate da CAN

Faixa [Ajuste fábrica] Unidade: 0...8 [0] bits/segundo

☑ Ajusta o baudrate da CAN. Os valores permitidos são:

P772	Descrição	Comprimento máximo do cabo
0	1 Mbit/s	25 m
1	Reservado	-
2	500 Kbit/s	100 m
3	250 Kbit/s	250 m
4	125 Kbit/s	500 m
5	100 Kbit/s	600 m
6	50 Kbit/s	1000 m
7	20 Kbit/s	1000 m
8	10 Kbit/s	1000 m

P773
Recuperar bus off

0...1 [0] -

☑ Permite selecionar a ação da PLC1 quando um erros de *bus off* ocorrer. Os valores permitidos são:

P773	Descrição	Observação
0	Manual	Após a detecção do error de *bus off*, o dispositivo indicará E61, a comunicação CAN será desabilitada, e o dispositivo deverá ser resetado manualmente para voltar a operar na rede.
1	Automático	A comunicação será reiniciada automaticamente após a detecção do erro de *bus off*.

P775
Status da CAN

0...5 [-] -

☑ Parâmetro de leitura.
☑ Informa o status da CAN, sendo:
0=Desabilitado
1=Reservado
2=CAN habilitado
3=*Warning* (alguns telegramas com erro)
4=*Error Passive* (muitos telegramas com erro, ou é o único dispositivo da rede com CAN habilitado e transmitindo telegramas).
5=*Bus off* (quantidade de erros detectados ultrapassou o limite interno do dispositivo, e a comunicação foi desabilitada).

P776
Contador de Telegramas Recebidos

0...65535 [-] -

☑ Parâmetro de leitura.
☑ Contador cíclico, incrementado a cada telegrama CAN recebido com sucesso. A contagem é reiniciada toda vez que o contador atinge o limite superior.

P777
Contador de Telegramas Transmitidos

0...65535 [-] -

☑ Parâmetro de leitura
☑ Contador cíclico, incrementado a cada telegrama CAN transmitido com sucesso. A contagem é reiniciada toda vez que o contador atinge o limite superior.

P778
Contador de Erros Detectados

0...65535 [-] -

☑ Parâmetro de leitura.
☑ Contador cíclico, incrementado a cada erro detectado (*warning*, *error passive* ou *bus off*). A contagem é reiniciada toda vez que o contador atinge o limite superior.

WLP

O WLP é o software para ambiente Windows, que serve para a programação do cartão PLC1 em linguagem Ladder. É facilmente instalável em um PC e a sua programação é simples.

Este manual descreve basicamente os procedimentos necessários para a instalação do WLP, a criação de projetos e fornece uma visão global dos blocos existentes na PLC1.

9.1 Instalação do WLP

1. Insira o disco na unidade de CD-ROM
2. Clique menu “Iniciar” e selecione o comando “Executar”.
3. Digite “d:setup.exe”.
   OBS:. Isto vale se o drive CD-ROM estiver no drive d:
4. Siga as instruções do Setup.

9.2 Iniciando a Programação

1. Abra o WLP.
2. Selecione a opção “Novo Projeto”.
3. Digite um nome para o projeto.
4. Inicie a programação utilizando os comandos da barra de edição.
5. Após o programa estar concluído, teclar <F7> (menu-construir-compilar) para efetuar a compilação do projeto e corrigir os erros, se necessário.
6. Conectar o cabo do PC a placa PLC.
7. Configurar a comunicação serial, selecionando a porta serial, o endereço da placa PLC na rede, a taxa de transmissão, teclando <Shift>+<F8> (menu-comunicação-configurações).
   OBS: A paridade deve ser sempre na opção “Sem Paridade”
8. Transmitir o programa teclando <F8> (menu-comunicação-transmitir programa do usuário).

9.3 Parâmetros Programáveis pelo Usuário

Permitem ao usuário uma boa flexibilidade na implementação dos projetos, pois podem ser facilmente acessados pela HMI do CFW-09. Consequentemente, o seu respectivo nome da função do parâmetro e sua unidade, podem ser editados no WLP através do editor de parâmetros (Alt+G), para posteriormente serem transmitidos as cartão PLC1.

9.4 Considerações Gerais dos Blocos Programáveis

9.4.1 Posição / Offset - SCURVE, TCURVE, HOME:

A posição / offset é composta por três partes:
- ☑ sinal
- ☑ número de voltas
- ☑ fração de voltas

Sinal:
O sinal é composto por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado do sinal pode ser:
- ☑ constante
- ☑ parâmetro do usuário
- ☑ marcador de bit
- ☑ entrada digital

Para o tipo de dado constante, o valor pode ser:
- ☑ positivo
- ☑ negativo

Número de Voltas:
O número de voltas é composto por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado pode ser:
- ☑ constante
- ☑ parâmetro do usuário
- ☑ marcador de word

Para o tipo de dado constante, o valor deve ser programado de acordo com a unidade configurada no projeto e o campo “Fração de Volta” não precisa ser configurado.

Para os parâmetros do usuário e os marcadores de word a unidade considerada por este campo é o número de rotações.

Fração de Volta:
A fração de volta é composta apenas por um endereço, pois ela compartilha do mesmo tipo de dado do campo “Número de Voltas”.

Se o tipo de dado for constante, este valor é ignorado, valendo apenas a constante configurada no campo “Número de Voltas”.

Para os parâmetros do usuário e os marcadores de word, a unidade considerada por este campo é número de pulsos, sendo que pode variar entre, 0 a 65535 pulsos, que equivale a uma faixa de 0 a 359,9945068359375º.

9.4.2 Velocidade - INBWG, SCURVE, TCURVE, HOME, JOG, SETSPEED:

A velocidade é composta por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado da velocidade pode ser:
- ☑ constante
- ☑ parâmetro do usuário
- ☑ marcador de word

Para o tipo de dado constante, o valor deve ser programado de acordo com a unidade configurada no projeto.

Para os parâmetros do usuário e os marcadores de word a unidade considerada por este campo é o RPM (rotações por minuto).

9.4.3 Aceleração/ Desaceleração - SCURVE, TCURVE, HOME, STOP, JOG, SETSPEED:

A aceleração/desaceleração é composta por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado da aceleração pode ser:
- ☑ constante
- ☑ parâmetro do usuário
- ☑ marcador de word

Para o tipo de dado constante, o valor deve ser programado de acordo com a unidade configurada no projeto.

Para os parâmetros do usuário e os marcadores de word a unidade considerada por este campo é RPM/s (rotações por minuto por segundo).

9.4.4 Jerk - SCURVE:

O jerk é composto por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado do jerk pode ser:
- ☑ constante
- ☑ parâmetro do usuário
- ☑ marcador de word

Para o tipo de dado constante, o valor deve ser programado de acordo com a unidade configurada no projeto.

Para os parâmetros do usuário e os marcadores de word a unidade considerada por este campo é RPM/s² (rotações por minuto por segundo ao quadrado).

9.4.5 Modo - SCURVE, TCURVE:

O modo é sempre uma constante.

Possui as opções:
- ☑ relativo
- ☑ absoluto

O modo relativo refere-se a um posicionamento a partir de sua última posição. Neste caso, o sentido de giro deste posicionamento é dado pelo sinal, ou seja, sentido horário se for positivo e sentido anti-horário se for negativo.

O modo absoluto refere-se a posição de zero máquina, só podendo ser utilizado se uma busca de zero já foi feita previamente.

9.4.6 Sentido de Rotação - INBWG, HOME, JOG, ETSPEED:

O sentido de rotação é sempre constante.

Possui as opções:
- ☑ horário
- ☑ anti-horário.

9.4.7 Parte Inteira - INT2FL, FL2INT:

A parte inteira é composta por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado da parte inteira pode ser:
- ☑ constante
- ☑ marcador de word
- ☑ parâmetro do usuário

ATENÇÃO!
Quando a parte inteira referir-se a um resultado de saída de qualquer bloco, o tipo de dado constante não é permitido.

9.4.8 Parte Fracionária - INT2FL, FL2INT:

A parte fracionária é composta por um tipo de dado e um endereço.

O tipo de dado da parte fracionária pode ser:
- ☑ constante
- ☑ marcador de word
- ☑ parâmetro do usuário

ATENÇÃO!
Quando a parte fracionária referir-se a um resultado de saída de qualquer bloco, o tipo de dado constante não é permitido.

9.4.9 Float - INT2FL, FL2INT, MATH, COMP, PID, SAT, FUNC, FILTER:

O float é composta por um tipo de dado e um endereço.

O tipo de dado do float pode ser:
- ☑ constante float
- ☑ marcador de float

ATENÇÃO!
Quando o float referir-se a um resultado de saída de qualquer bloco, o tipo de dado constante float não é permitido.

Os limites do float são:
- ☑ representação máxima = 3.402823466e+38F
- ☑ valor mínimo positivo = 1.175494351e–38F

9.4.10 Limites - PID, SAT:

Os limites são compostos por 2 partes:
- ☑ float – máximo (ver item 9.4.9)
- ☑ float – mínimo (ver item 9.4.9)

9.4.11 Valores de Entrada / Valores de Saída - SAT, FUNC, FILTER:

Os valores são compostos por 2 partes:
- ☑ float – entrada (ver item 9.4.9)
- ☑ float – saída (ver item 9.4.9)

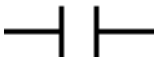
9.5 Faixa de Endereços

Tipo	Faixa	Quantidade
Marcadores de Bit Retentivos	%MX1000 ... %MX1671	672
Marcadores de Bit Voláteis	%MX2000 ... %MX3407	1308
Marcadores de Word Retentivos	%MW6000 ... %MW6299	300
Marcadores de Word Voláteis	%MW7000 ... %MW7799	800
Marcador de sistema	%SW0	1
Marcadores de Float Voláteis	%MF9000 ... % MF9099	100
Parâmetros do Usuário	%UW800 ... %UW899	100
Entradas Digitais	%IX1 ... %IX9	9
Entradas Digitais do Drive	%IX101 ... %IX106	6
Saídas Digitais	%QX1 ... %QX6	6
Saídas Digitais do Drive	%QX101 ... %QX103	3
Entradas Analógicas do Drive	%IW101 ... %IW102	2
Saídas Analógicas do Drive	%QW101 ... %QW102,	2

BLOCOS DO WLP

10.1 Contato Normalmente Aberto (NOCONTACT)

Figura:



Descrição:
É composto por 1 entrada, 1 saída e 1 argumento.

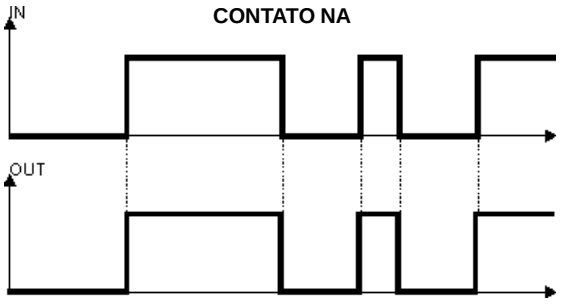
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ entrada digital
- ☑ saída digital
- ☑ parâmetro do usuário

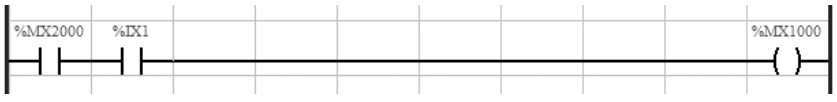
NOTA!
Na opção parâmetro do usuário, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Transfere o sinal contido em sua entrada para a sua saída, se o valor do seu argumento for 1. Caso contrário, transfere 0 para a sua saída.

Gráfico:



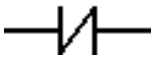
Exemplo:



Se o marcador de bit 2000 e a entrada digital 1 forem 1, escreve 1 no marcador de bit 1000. Caso contrário, escreve 0.

10.2 Contato Normalmente Fechado (NCCONTACT)

Figura:



Descrição:
É composto por 1 entrada, 1 saída e 1 argumento.

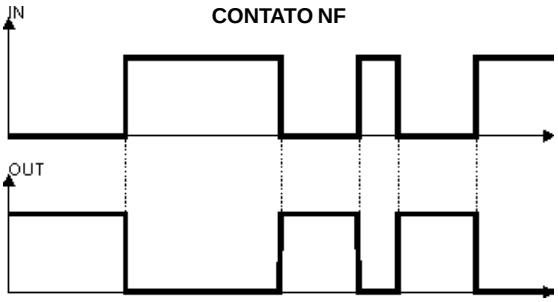
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ entrada digital
- ☑ saída digital
- ☑ parâmetro do usuário

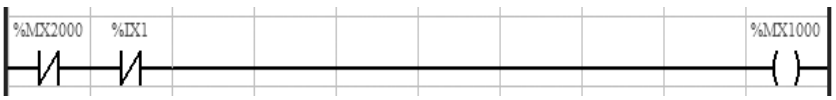
NOTA!
Na opção parâmetro do usuário, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Transfere o sinal contido em sua entrada para a sua saída, se o valor do seu argumento for 0. Caso contrário, transfere 0 para a sua saída.

Gráfico:



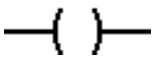
Exemplo:



Se o marcador de bit 2000 e a entrada digital 1 forem 0, escreve 1 no marcador de bit 1000. Caso contrário, escreve 0.

10.3 Bobina (COIL)

Figura:



Descrição:
É composto por 1 entrada e 1 argumento.

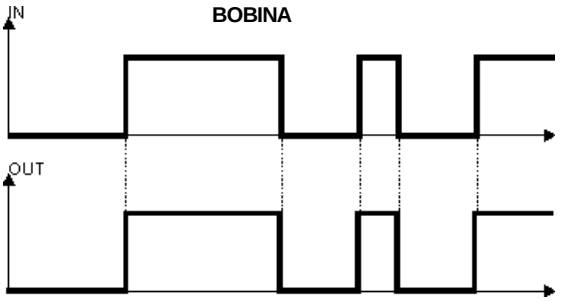
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ saída digital
- ☑ parâmetro do usuário

NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado. Além disso, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Transfere o sinal contido em sua entrada para o seu argumento.

Gráfico:



Exemplo:

Se o marcador de bit 2000 ou a entrada digital 1 for 1, escreve 1 no marcador de bit 1000. Caso contrário, escreve 0.

10.4 Bobina Negada (NEGCOIL)

Figura:

Descrição:
É composto por 1 entrada e 1 argumento.

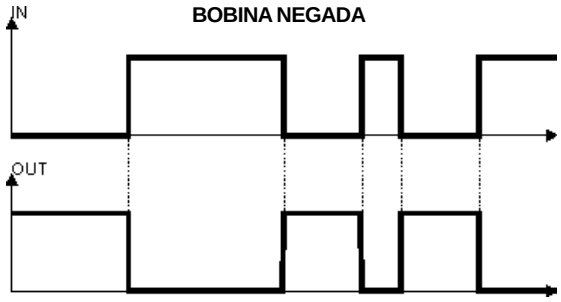
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ saída digital
- ☑ parâmetro do usuário

NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado. Além disso, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Transfere o inverso do sinal contido em sua entrada para o seu argumento.

Gráfico:



Exemplo:

Se o marcador de bit 2000 ou a entrada digital 1 for 1, e o parâmetro do usuário 800 for 0, escreve 0 na saída digital 1. Caso contrário, escreve 1.

10.5 Seta Bobina (SETCOIL)

Figura:

Descrição:
É composto por 1 entrada e 1 argumento.

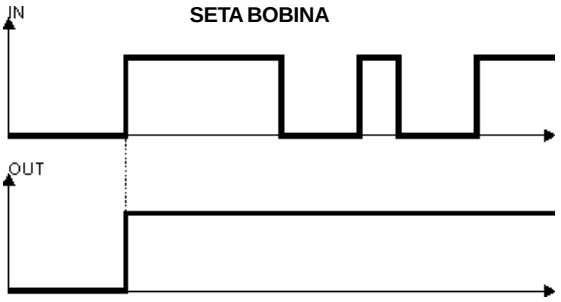
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ saída digital
- ☑ parâmetro do usuário

NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado. Além disso, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Quando o sinal de entrada for 1, o argumento é setado. O argumento somente será resetado quando um componente reseta bobina for ativado.

Gráfico:

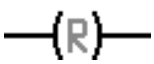


Exemplo:

Se o parâmetro do usuário 801 e a saída digital 1 do drive forem 1, ou a entrada digital 1 for 1, e o parâmetro do usuário 800 for 0, seta a saída digital 1. Caso contrário, o valor da saída é mantido.

10.6 Reseta Bobina (RESETCOIL)

Figura:



Descrição:
É composto por 1 entrada e 1 argumento.

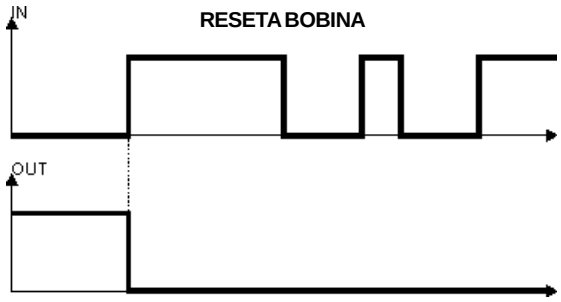
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ saída digital
- ☑ parâmetro do usuário

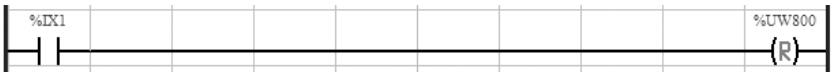
NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado. Além disso, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Quando o sinal de entrada for 1, o argumento é resetado. O argumento somente será setado quando um componente seta bobina for ativado.

Gráfico:



Exemplo:



Se a entrada digital 1 for 1, reseta o parâmetro do usuário 800. Caso contrário, o valor do parâmetro é mantido.

10.7 Bobina de Transição Positiva (PTSCOIL)

Figura:

Descrição:
É composto por 1 entrada e 1 argumento.

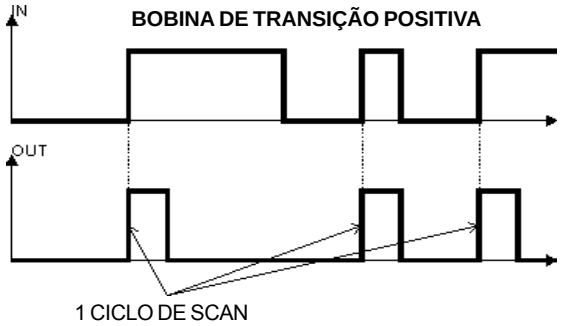
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ saída digital
- ☑ parâmetro do usuário

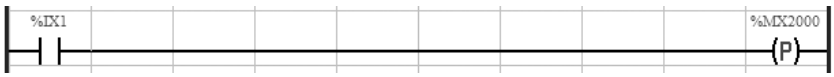
NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado. Além disso, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Quando houver uma transição de 0 para 1 no sinal de entrada, o argumento é setado durante um ciclo de scan. Depois disso o argumento é resetado, mesmo que a sua entrada permaneça em 1.

Gráfico:



Exemplo:



Quando a entrada digital 1 for de 0 para 1, escreve 1 por um ciclo de scan no marcador de bit 2000.

10.8 Bobina de Transição Negativa (NTSCOIL)

Figura:

Descrição:
É composto por 1 entrada e 1 argumento.

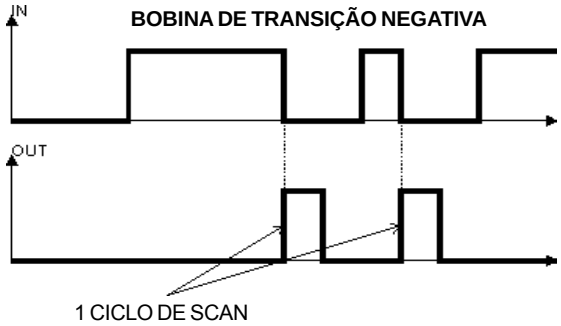
O argumento é composto por um tipo de dado e um endereço.

O tipo de dado do argumento pode ser:
- ☑ marcador de bit
- ☑ saída digital
- ☑ parâmetro do usuário

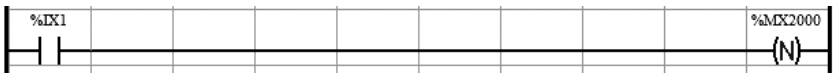
NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado. Além disso, valores pares correspondem a 0, enquanto que valores ímpares correspondem a 1.

Funcionamento:
Quando houver uma transição de 1 para 0 no sinal de entrada, o argumento é setado durante um ciclo de scan. Depois disso, o argumento é resetado, mesmo que a sua entrada permaneça em 0.

Gráfico:



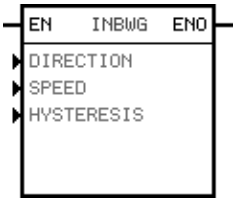
Exemplo:



Quando a entrada digital 1 for de 1 para 0, escreve 1 por um ciclo de scan no marcador de bit 2000.

10.9 Bloco Em Movimento (INBWG)

Figura:



Descrição:
É composto por 1 entrada EN, 1 saída ENO e 2 argumentos, sendo eles:
- ☑ velocidade (ver item 9.4.2)
- ☑ sentido de rotação (ver item 9.4.6)
- ☑ histerese

A entrada EN é responsável pela habilitação do bloco.

A saída ENO informa se o sentido de rotação é o mesmo do sentido programado e se a velocidade do motor é maior ou igual ao valor programado.

Funcionamento:
Se a entrada EN for 0, o bloco não é executado e a saída ENO vai para 0.

Se a entrada EN for 1, o bloco compara a velocidade e o sentido de giro do motor com os argumentos de velocidade, sentido de giro e histerese programados.

Se o motor está girando no mesmo sentido do argumento de sentido de rotação programado e a velocidade do motor for maior ou igual do que o argumento de velocidade programado mais a histerese, então é transferido 1 para a saída ENO. Caso contrário, é transferido 0 para a saída ENO.

A histerese serve para impedir oscilação na saída do bloco quando a velocidade real está muito próxima ou igual à velocidade programada. Por exemplo, para uma velocidade de 1000rpm com histerese de 1%, o bloco liga a saída ENO quando a velocidade real atingir 1010rpm, e somente a desliga quando a velocidade cair em torno dos 990rpm. A histerese é dada em porcentagem, podendo variar entre 0.0% e 50.0%. Se for programada por parâmetro a unidade passa a ser "por mil", variando de 0 a 500.

Fluxograma:

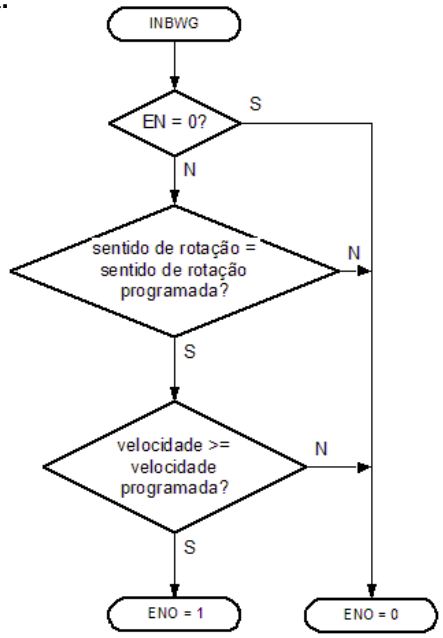
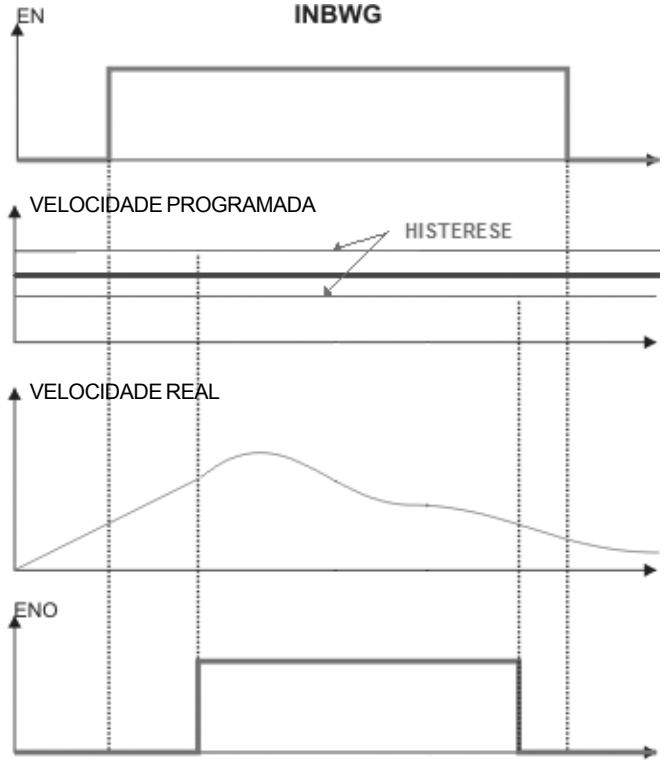
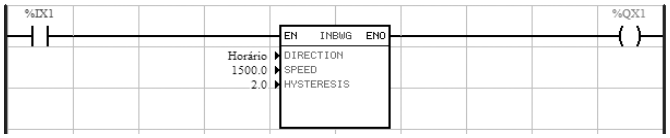


Gráfico:



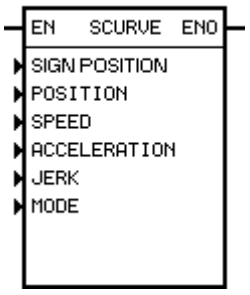
Exemplo:



Enquanto a entrada digital 1 estiver em 1, o bloco INBWG está ativado. Neste caso, se a o motor estiver rodando no sentido horário e sua velocidade for maior ou igual a 1500 rpm (respeitando a histerese de 2%), escreve 1 na saída digital 1. Caso contrário, escreve 0.

10.10 Bloco Curva S (SCURVE)

Figura:



Descrição:
É composto por 1 entrada EN, 1 saída ENO e 5 argumentos, sendo eles:
- ☑ posição (sinal, número de voltas, fração de volta) (ver item 9.4.1)
- ☑ velocidade (ver item 9.4.2)
- ☑ aceleração (ver item 9.4.3)
- ☑ jerk (ver item 9.4.4)
- ☑ modo (ver item 9.4.5)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO informa o instante em que o bloco é finalizado.

Funcionamento:
Se a entrada EN for 0, o bloco não é executado e a saída ENO vai para 0.

Se houver pelo menos um pulso durante um ciclo de scan na entrada EN e não houver outro bloco de posicionamento ativo, será executado um posicionamento com um perfil S baseado nas características programadas nos argumentos.

Quando o posicionamento termina, a saída ENO vai para 1 durante um ciclo de scan, retornando posteriormente a 0.

Importante: Este bloco trabalha em malha de posição, permanecendo assim mesmo após a sua conclusão.

Fluxograma:

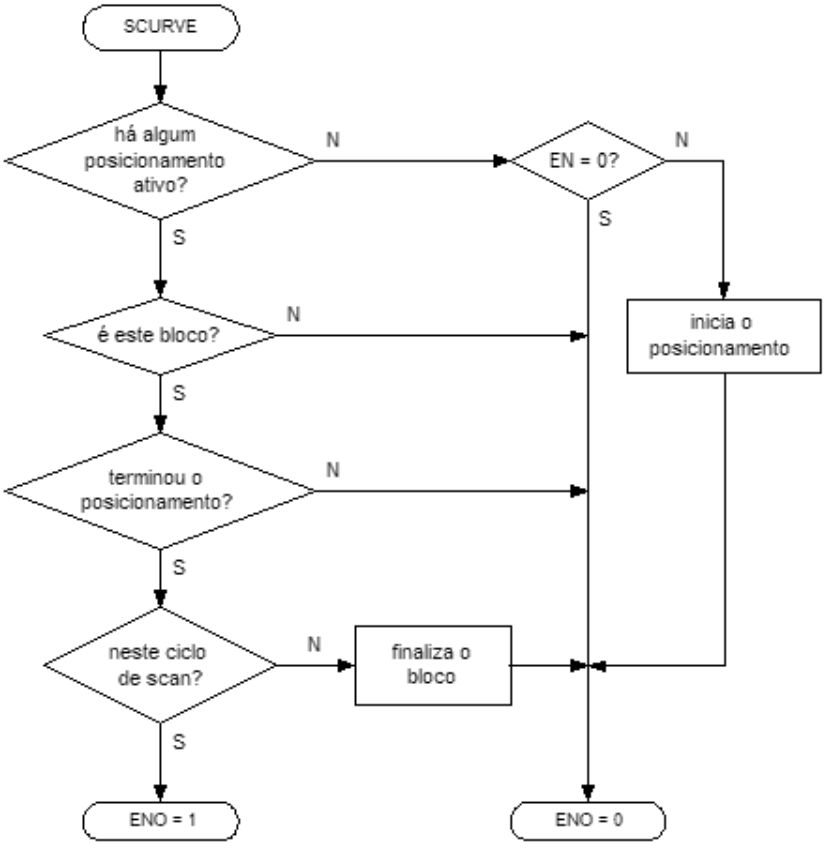
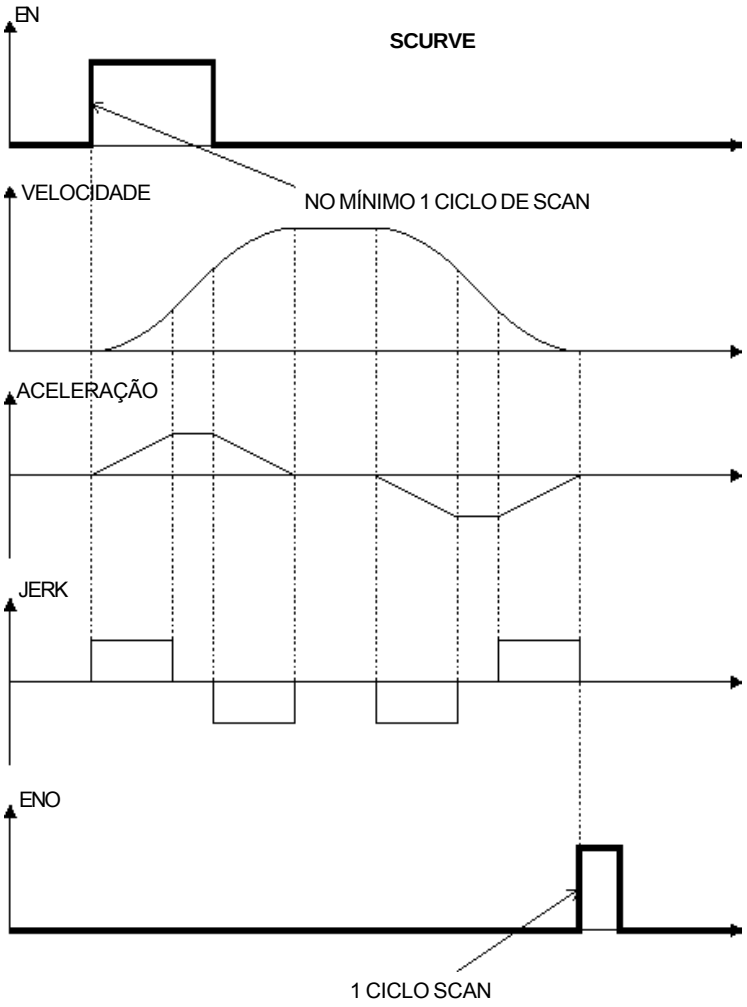


Gráfico:



Equações da cinemática que regem este posicionamento:

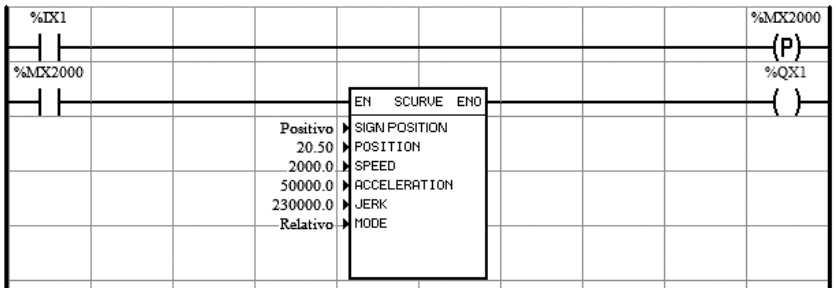
$$x = x_0 + v_0 \cdot t + \frac{1}{2} \cdot a_0 \cdot t^2 + \frac{1}{6} \cdot J \cdot t^3$$

$$v = v_0 + a_0 \cdot t + \frac{1}{2} \cdot J \cdot t^2$$

$$a = a_0 + J \cdot t$$

- x = posição final
- x_0 = posição inicial
- v = velocidade final
- v_0 = velocidade inicial
- a = aceleração final
- a_0 = aceleração inicial
- J = jerk

Exemplo:

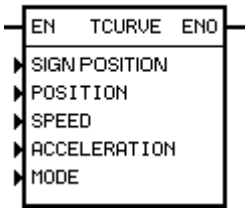


Quando for capturado uma transição de 0 para 1 na entrada digital 1, dispara um posicionamento de 20,5 voltas, a uma velocidade de 2000 rpm, com uma aceleração de 50000 rpm/s e um jerk de 230000 rpm/s², no sentido horário, pois o modo é relativo e o sinal da posição é positivo. Quando o posicionamento terminar, escreve 1 durante 1 ciclo de scan na saída digital 1.

Lembramos que o jerk é a derivada da aceleração em função do tempo. Desta forma, conclui-se que a aceleração máxima será atingida em 50000 rpm/s / 230000 rpm/s² = 0,22 segundos.

10.11 Bloco Curva Trapezoidal (TCURVE)

Figura:



Descrição:
É composto por 1 entrada EN, 1 saída ENO e 4 argumentos, sendo eles:
- ☑ posição (sinal, número de voltas, fração de volta) (ver item 9.4.1)
- ☑ velocidade (ver item 9.4.2)
- ☑ aceleração (ver item 9.4.3)
- ☑ modo (ver item 9.4.5)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO informa o instante em que o bloco é finalizado.

Funcionamento:
Se a entrada EN for 0, o bloco não é executado e a saída ENO vai para 0.

Se houver pelo menos um pulso durante um ciclo de scan na entrada EN e não houver outro bloco de posicionamento ativo, será executado um posicionamento com um perfil trapezoidal baseado nas características programadas nos argumentos.

Quando o posicionamento termina, a saída ENO vai para 1 durante um ciclo de scan, retornando posteriormente a 0.

Importante: Este bloco trabalha em malha de posição, permanecendo assim mesmo após a sua conclusão.

Fluxograma:

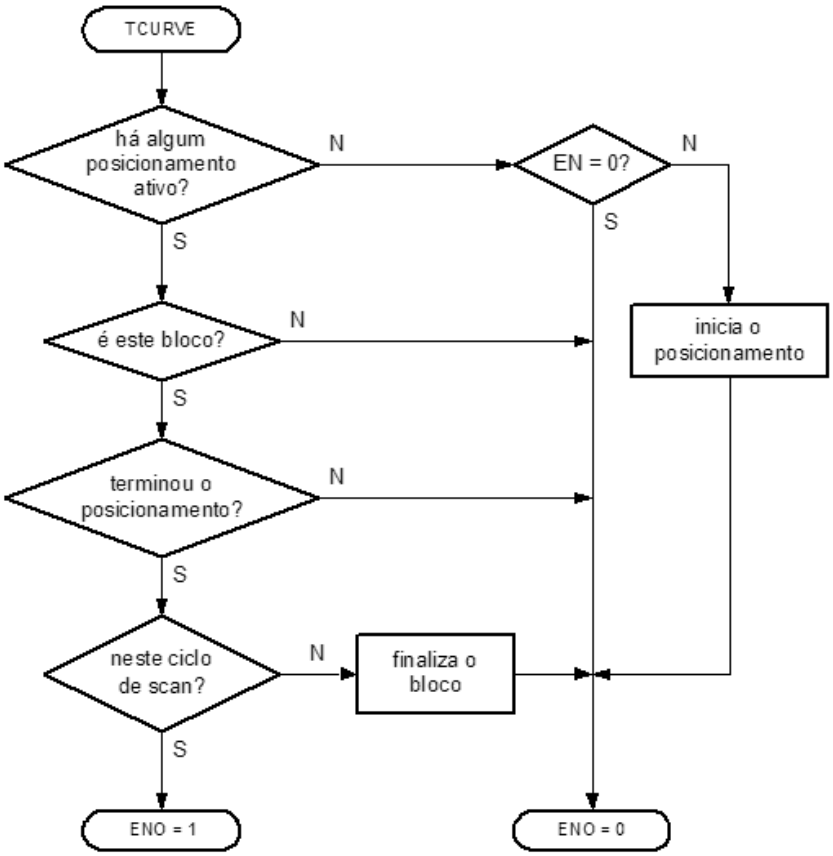
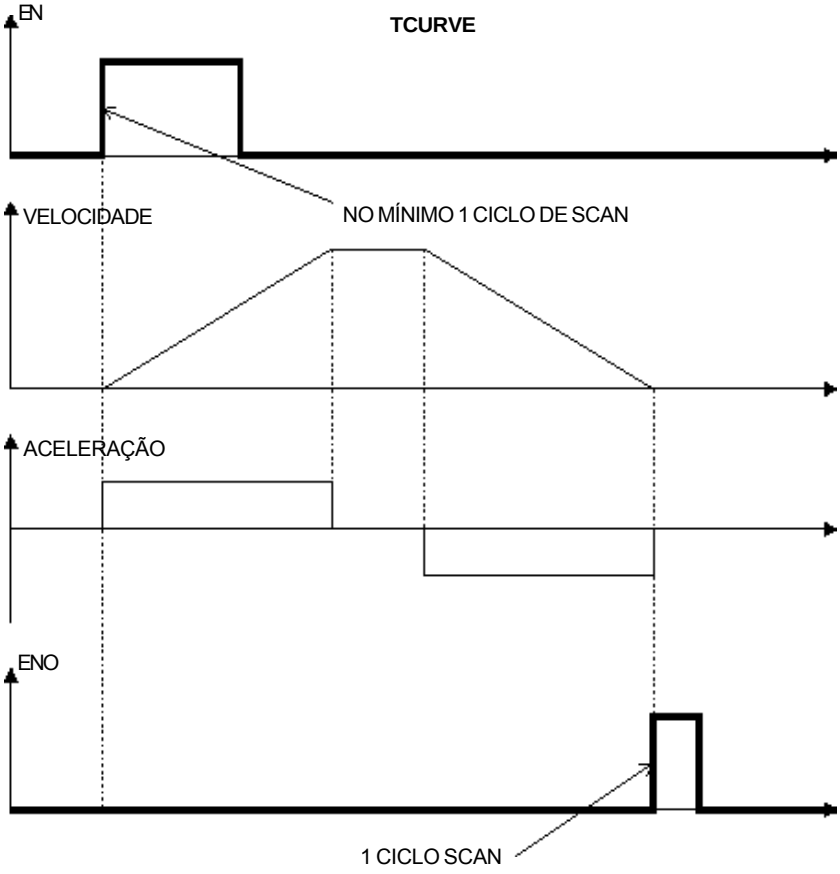


Gráfico:



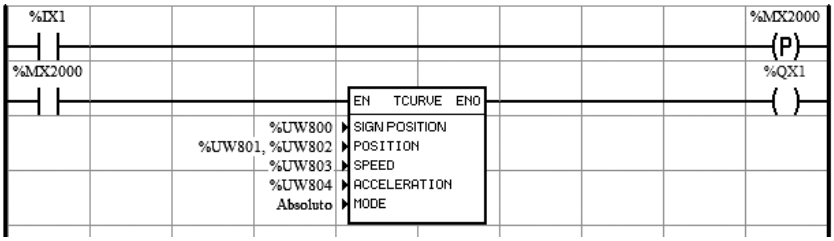
Equações da cinemática que regem este posicionamento:

$$x = x_0 + v_0 \cdot t + \frac{1}{2} \cdot a_0 \cdot t^2$$

$$v = v_0 + a_0 \cdot t$$

onde:
- x = posição final
- x_0 = posição inicial
- v = velocidade final
- v_0 = velocidade inicial
- a = aceleração final

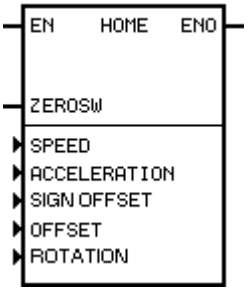
Exemplo:



Quando for capturado uma transição de 0 para 1 na entrada digital 1, dispara um posicionamento para a posição absoluta configurada com sinal do parâmetro do usuário 800, com o número de voltas do parâmetro do usuário 801 e com a fração de volta do parâmetro do usuário 802, na velocidade do parâmetro do usuário 803 em rpm e com uma aceleração baseada no parâmetro do usuário 904 em rpm/s. Para isto é necessário que uma busca de zero máquina já tenha sido executada previamente. Quando terminar, escreve 1 durante 1 ciclo de scan na saída digital 1.

10.12 Bloco Busca de Zero Máquina (HOME)

Figura:



Descrição:
É composto por 1 entrada EN, 1 entrada ZEROSW, 1 saída ENO e 4 argumentos, sendo eles:
- ☑ sentido de rotação (ver item 9.4.6)
- ☑ velocidade (ver item 9.4.2)
- ☑ aceleração (ver item 9.4.3)
- ☑ offset (sinal, número de voltas, fração de volta) (ver item 9.4.1)

A entrada EN é responsável pela habilitação do bloco.

A entrada ZEROSW é responsável de informar ao bloco que a posição de zero máquina foi atingida.

A saída ENO informa o instante que o bloco é finalizado.

Funcionamento:

Se a entrada EN for 0, o bloco não é executado e a saída ENO fica em 0.

Se houver pelo menos um pulso durante um ciclo de scan na entrada EN e não haver outro bloco de posicionamento ativo, a busca de zero é iniciada com um perfil trapezoidal baseado nas características programadas nos argumentos.

No instante em que houver um pulso de no mínimo um ciclo de scan na entrada ZEROSW, inicia-se a busca do pulso nulo. Assim que o pulso nulo for encontrado, inicia-se o processo de parada seguido do retorno a posição do pulso nulo.

Então o bloco é finalizado e a saída ENO vai para 1 por um ciclo de scan, retornando a 0 posteriormente.

NOTA!
Na hipóstese deste bloco ser habilitado e a entrada ZEROSW estar em 1, a busca se inicia no sentido oposto ao programado até a entrada ZEROSW ir para 0. Neste instante, o bloco inverte o sentido de giro, repetindo o passo descrito no parágrafo anterior.

Na finalização deste bloco, a posição encontrada será referenciada com o valor do offset programado, que normalmente possui o valor zero. Se programássemos um offset negativo de 25 rotações, e executássemos um posicionamento relativo de 50 rotações com sinal positivo, a posição alcançada seria de 25 voltas e 0 de fração de volta, com sinal positivo. No entanto, se o posicionamento fosse absoluto, a posição final é 50 voltas e de 0 de fração de volta, com sinal positivo, girando na realidade 75 voltas no sentido horário.

NOTA!
A posição final pode sofrer um offset dependendo do valor do parâmetro 769, que provoca um adiantamento da posição em relação ao pulso nulo. Desta forma, a parada será o valor de P769 décimos de graus antes do pulso nulo.

ATENÇÃO!
Após a busca de zero de máquina, o controle fica em malha de posição.

Fluxograma:

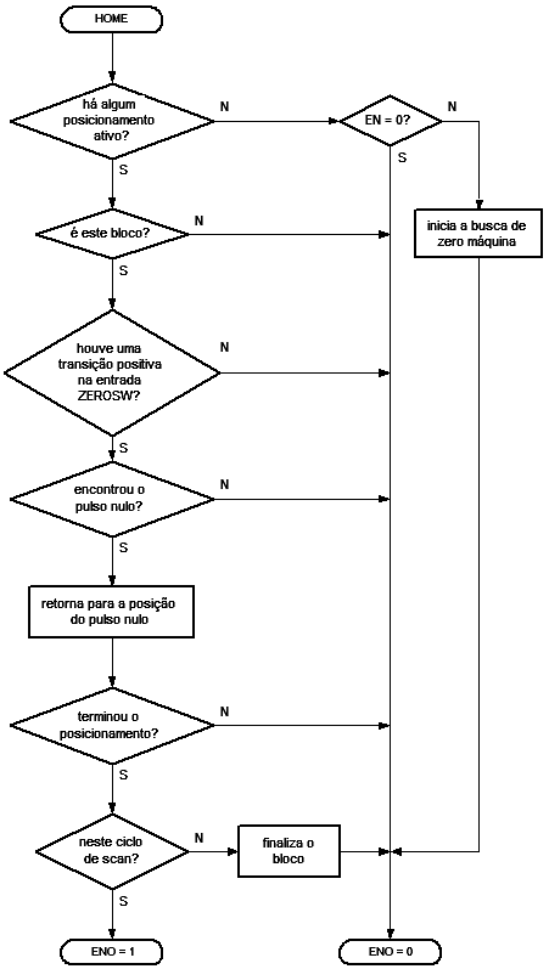
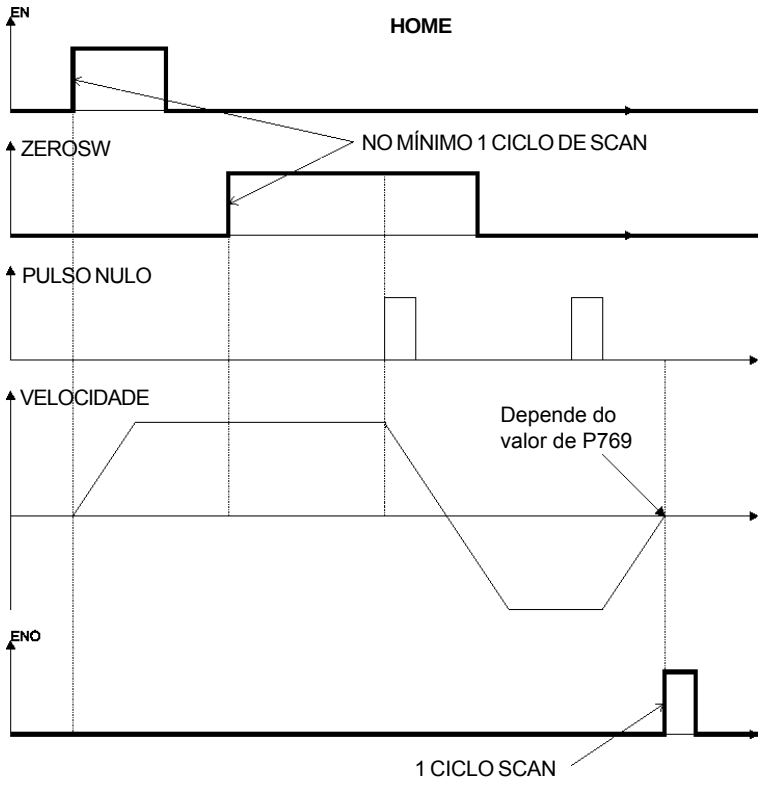
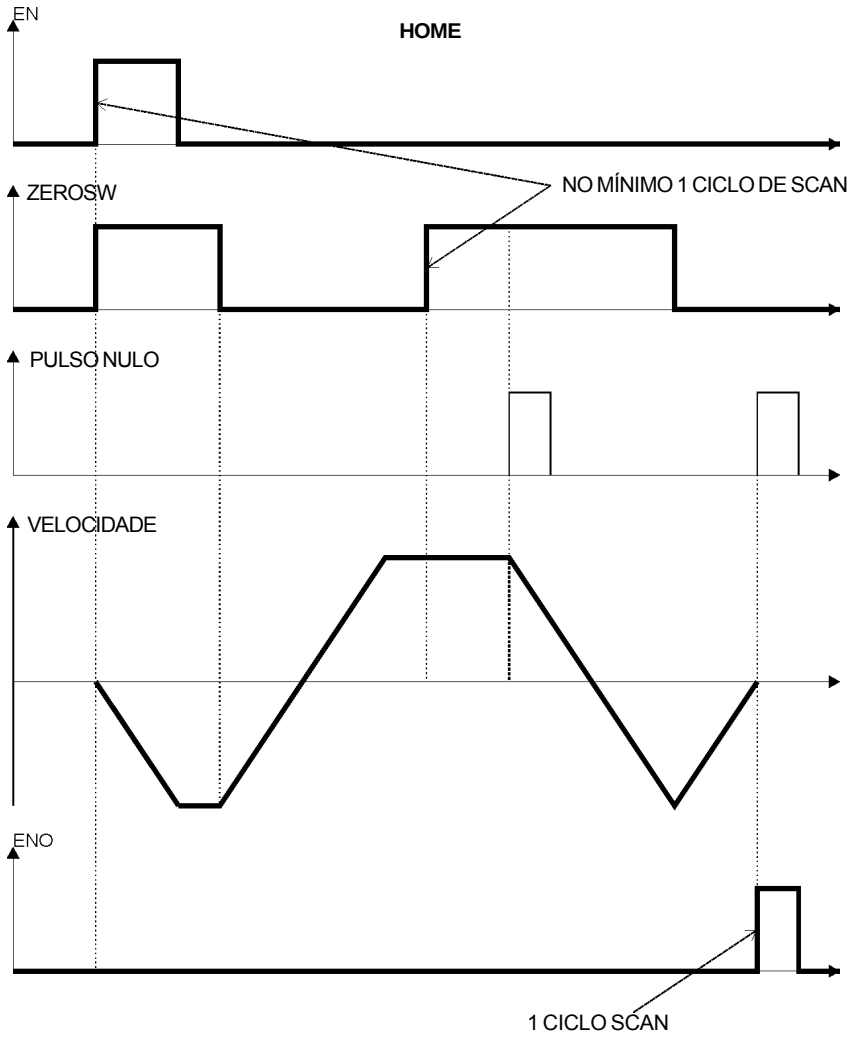


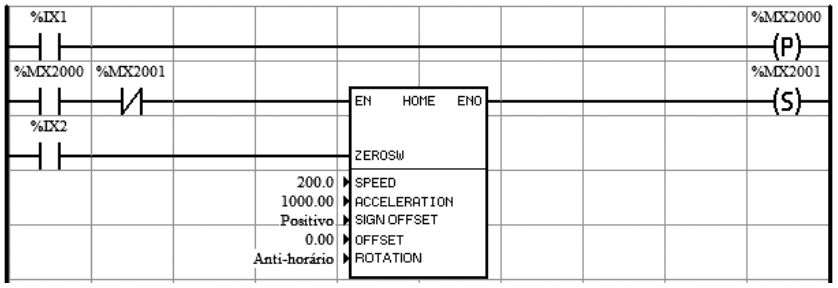
Gráfico:
Condição Normal - ZEROSW = 0



Exceção - ZEROSW = 1



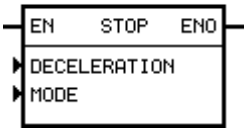
Exemplo:



Considerando que o drive tenha sido recém resetado ou energizado, na transição de 0 para 1 da entrada digital 1, ativa a busca de zero máquina, pois o marcador de bit 2001 é inicializado em 0. Quando a entrada 2 vai para 1, inicia a busca do pulso nulo. Ao encontrá-lo, o motor começa a desacelerar e volta para a posição do pulso nulo encontrada mais o valor de P769. Assim que o posicionamento é concluído, o marcador 2001 é setado, que inabilita uma nova busca.

10.13 Bloco Parada (STOP)

Figura:



Descrição:
É composto por 1 entrada EN, 1 saída ENO e 2 argumentos, sendo eles:
☑ desaceleração (ver item 9.4.3)
☑ modo

A entrada EN é responsável pela habilitação do bloco.

A saída ENO informa o instante que o bloco é finalizado.

Modo:
O modo é sempre uma constante.

Possui as opções:
☑ interrompe
☑ cancela

Funcionamento:
Se a entrada EN for 0, o este bloco não está ativo, a saída ENO fica em 0.

Se a entrada EN for 1, mesmo que seja por um ciclo de scan, é executado uma parada com um perfil trapezoidal baseado nas características programadas nos argumentos.

Quando a parada é concluída, a saída ENO vai para 1 durante um ciclo de scan, retornando a 0 posteriormente.

Depois de iniciado, o bloco de parada não é mais cancelado até a sua parada total, mesmo que a entrada EN vá para 0 antes do fim de sua parada.

O modo interrompe faz com que o bloco permaneça parado enquanto a entrada EN for 1. No instante que a entrada EN for 0, o bloco de posicionamento previamente ativo é restaurado, desde que a posição corrente não seja maior ou igual a posição desejada pelo posicionamento previamente ativo. Isto poderia ocorrer, se a desaceleração do bloco de parada fosse muito lento.

O modo cancela não restaura o posicionamento prévio quando a entrada EN for 0.

Nota: Se utilizado para parar uma busca de zero máquina, o modo de parada sempre será cancela, mesmo que a programação esteje setada para interrompe.

Importante: Este bloco não altera a forma de controle, seja ela em malha de posição ou em malha de velocidade.

Fluxograma:

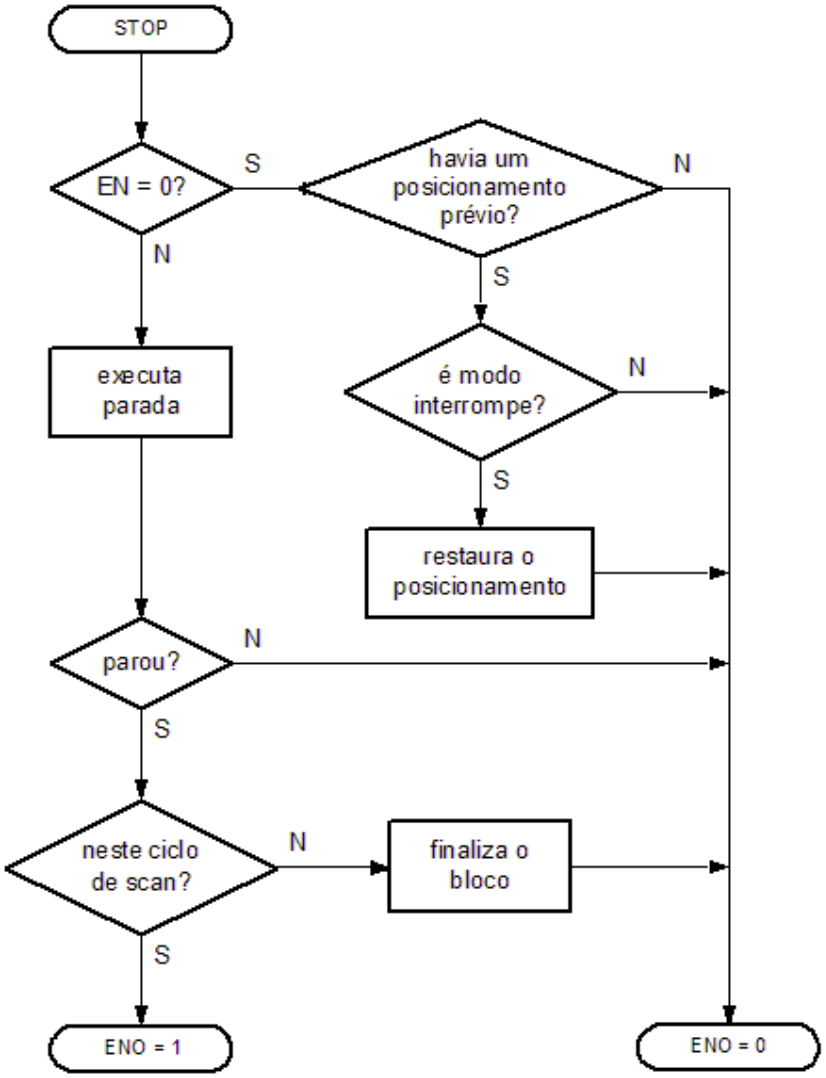
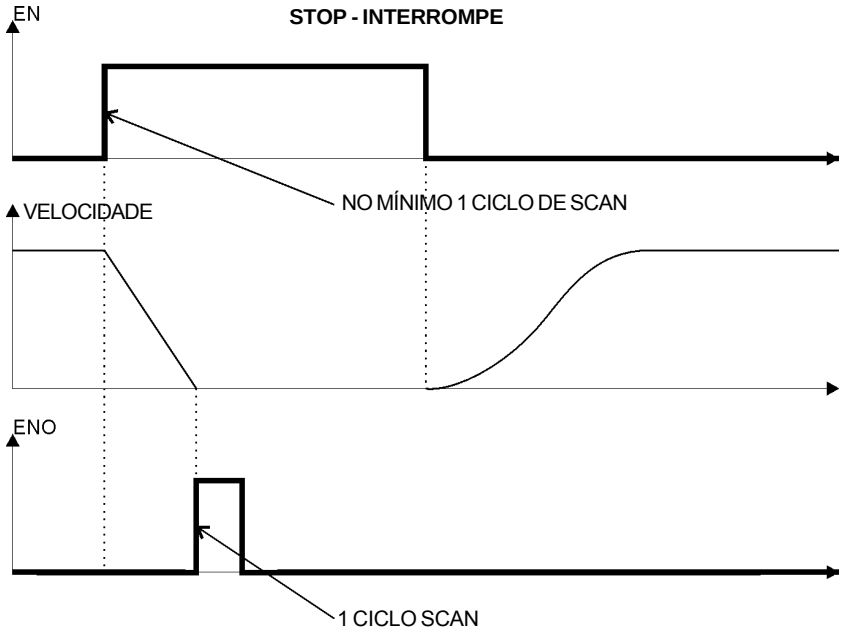
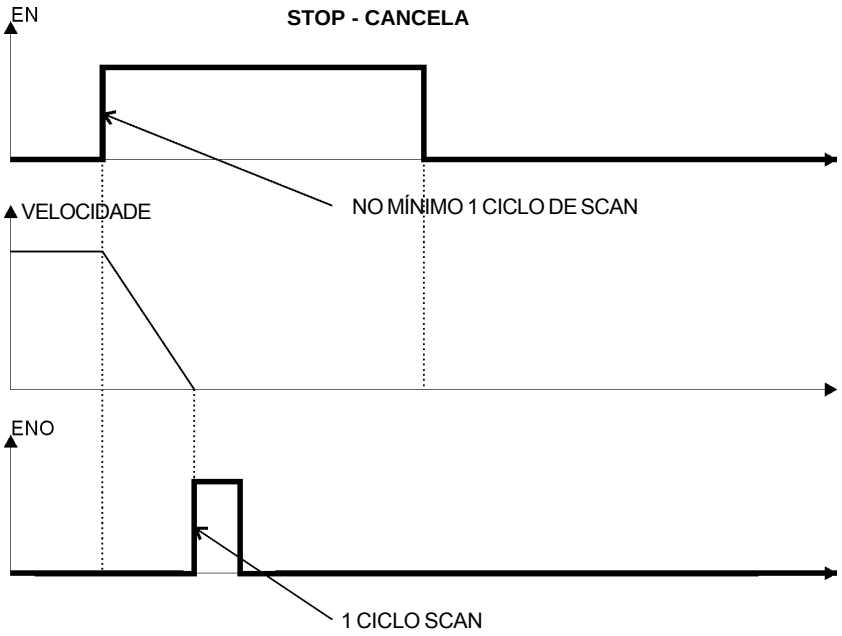


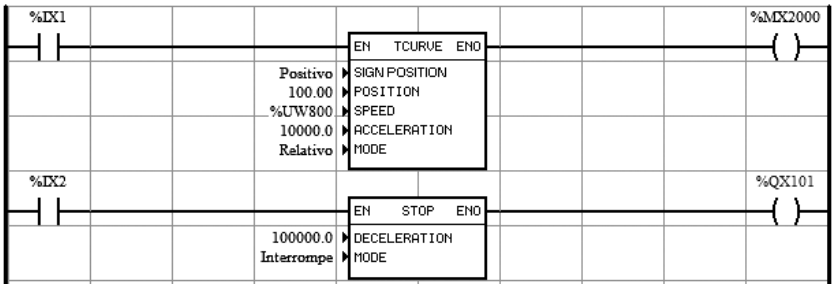
Gráfico:



Observe que para este caso, após a entrada EN ir para 0, uma curva S é inicializada, pois ela estava sendo executada antes de ocorrer uma parada.



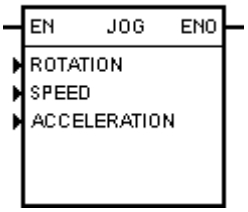
Exemplo:



Quando a entrada digital 1 for 1, um posicionamento de 100 voltas é habilitado. Se a entrada digital 2 for 1, o bloco de parada é habilitado, fazendo com que o posicionamento seje interrompido. Ao parar, é escrito na saída digital 1 do drive 1 por um ciclo de scan. No instante que a entrada digital 2 voltar para 0, o posicionamento de 100 voltas é completado.

10.14 Bloco Jog (JOG)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 3 argumentos, sendo eles:
- ☑ sentido de rotação (ver item 9.4.6)
- ☑ velocidade (ver item 9.4.2)
- ☑ aceleração (ver item 9.4.3)

A entrada EN é responsável pela habilitação do bloco.

Funcionamento:

Se a entrada EN for 0, o bloco não é executado e a saída ENO fica em 0.

Se a entrada EN for 1 e nenhum outro bloco de posicionamento estiver ativo, o bloco executa um perfil trapezoidal baseado nas características programadas nos argumentos e inicia a desaceleração quando a entrada EN for 0.

No instante que a entrada EN for para 0, inicia-se a parada e quando ela for finalizada, a saída ENO vai para 1 por um ciclo de scan, retornando a 0 posteriormente.

NOTA!
A velocidade do JOG não é atualizada online, ou seja, mesmo que o valor da velocidade programada seja alterada, a velocidade deste bloco não sofrerá alteração.

Importante: Este bloco trabalha em malha de velocidade, permanecendo assim mesmo após a sua conclusão.

Fluxograma:

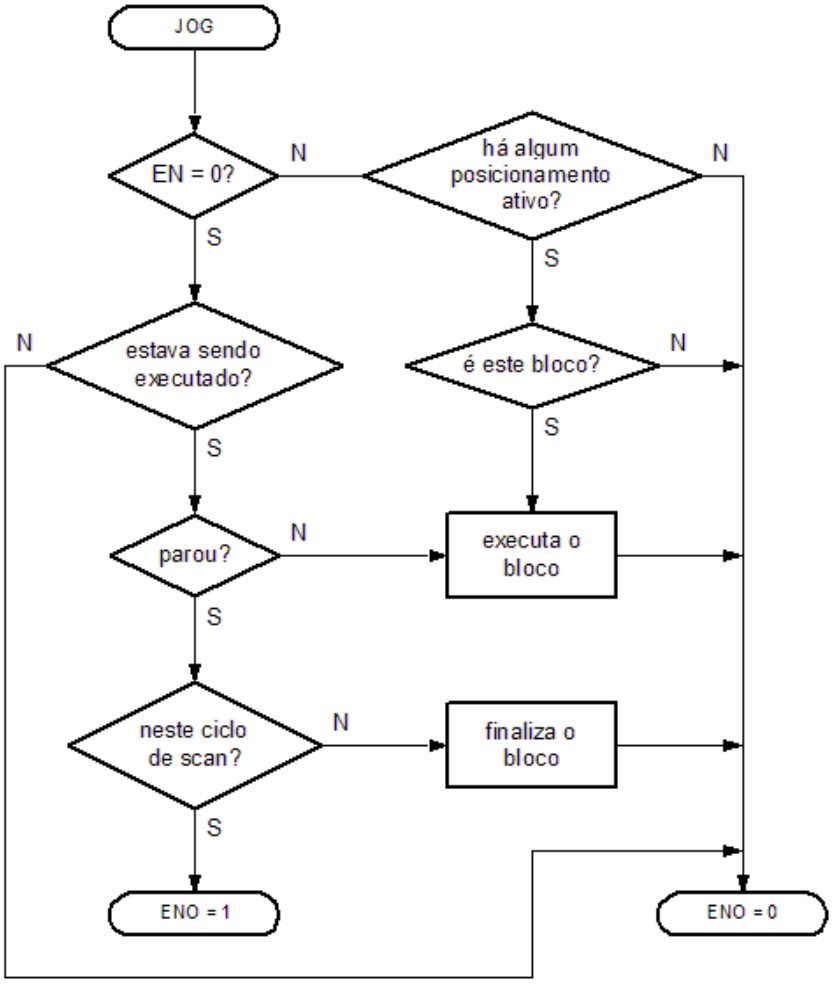
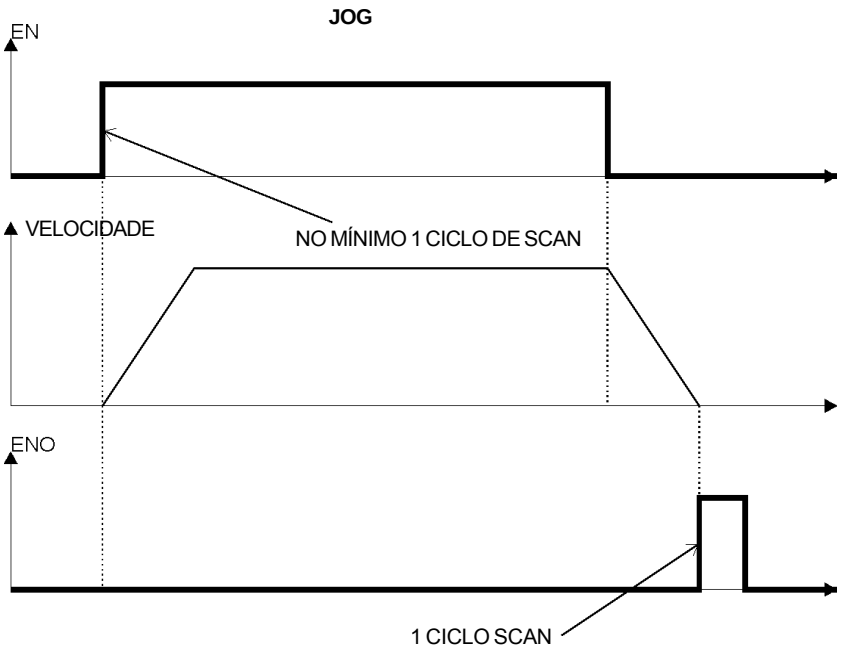
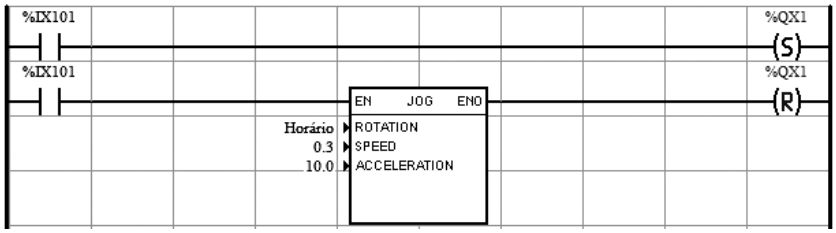


Gráfico:



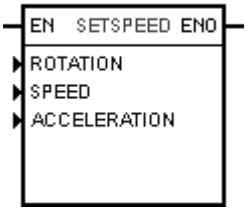
Exemplo:



Quando a entrada digital 1 do drive for 1, a saída digital 1 é setada e ao mesmo tempo o JOG é habilitado com uma velocidade de 0,3 rps. Quando a entrada 1voltar para 0, no momento que o bloco termina, ou seja, para totalmente, a saída 1 é resetada.

10.15 Bloco Seta Velocidade (Set Speed)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 3 argumentos, sendo eles:
- ☑ sentido de rotação (ver item 9.4.6)
- ☑ velocidade (ver item 9.4.2)
- ☑ aceleração (ver item 9.4.3)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO informa quando a velocidade do motor atingir a velocidade programada.

Funcionamento:

Se a entrada EN for 0, o bloco não é executado e saída ENO fica em 0.

Se a entrada EN sofrer uma transição de 0 para 1 e nenhum outro bloco de movimento estiver ativo, com exceção do próprio bloco Seta Velocidade, é executado um perfil trapezoidal baseado nas características programadas dos argumentos e nunca é finalizado. No entanto, outros blocos Seta Velocidade podem ser habilitados online, alterando a programação dos seus argumentos.

Para acabar com este movimento é necessário utilizar o bloco parada.

A saída ENO só vai para 1 em um ciclo de scan, quando o bloco atingir a velocidade programada. Caso contrário sempre é 0.

Importante: Este bloco trabalha em malha de velocidade, permanecendo assim mesmo após a sua conclusão.

Fluxograma:

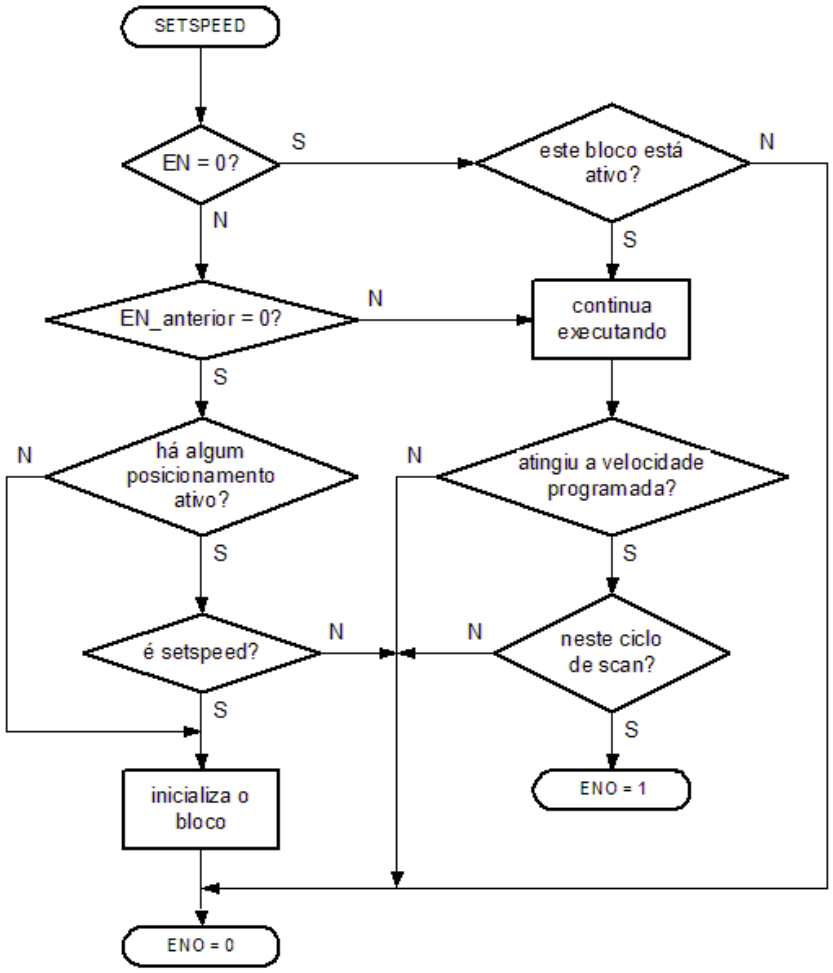
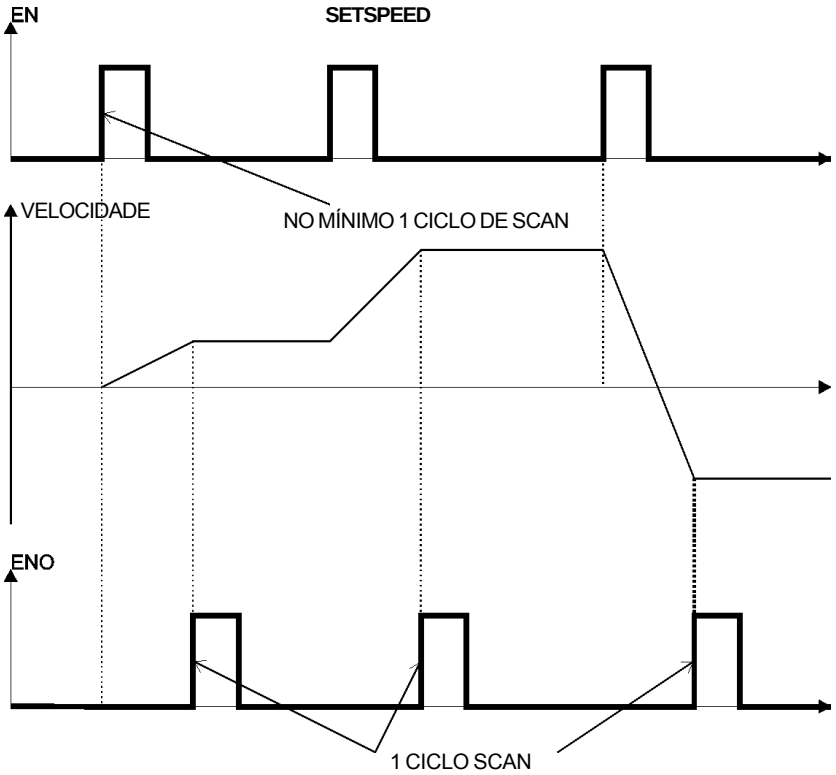
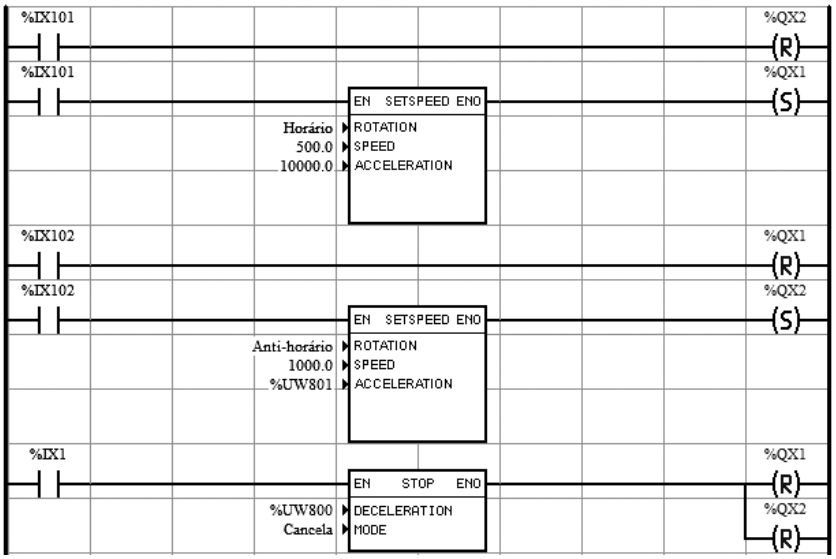


Gráfico:



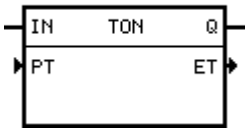
Exemplo:



Na transição de 0 para 1 da entrada digital 1 do drive, o bloco com velocidade de 500 rpm no sentido horário é disparado. Quando esta velocidade é atingida, a saída digital 1 é setada. Na transição de 0 para 1 da entrada digital 2 do drive, o bloco com velocidade de 1000 rpm no sentido anti-horário é disparado e a saída digital 1 é resetada. Quando esta nova velocidade é atingida, a saída digital 2 é setada. Se a entrada digital 1 for acionada, qualquer um dos dois movimentos prévios que está ativo é cancelado e o motor para, e ambas saídas 1 e 2 são resetadas."

10.16 Bloco Temporizador (TON)

Figura:



Descrição:
É composto por 1 entrada IN, 1 saída Q e 2 argumentos, sendo eles:
- ☑ PT – tempo desejado
- ☑ ET – tempo decorrido

A entrada IN é responsável pela habilitação do bloco.

A saída Q informa se o tempo decorrido atingiu o tempo programado.

PT (Tempo Desejado):

O tempo desejado é composto por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado do sinal pode ser:
- ☑ constante
- ☑ parâmetro do usuário
- ☑ marcador de word

Para o tipo de dado constante, o valor máximo permitido é 30.000 ms.

ET (Tempo Decorrido):

O tempo decorrido é composto por um tipo de dado e um endereço.

O tipo de dado do tempo decorrido pode ser:
- ☑ parâmetro do usuário
- ☑ marcador de word

NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado.

Funcionamento:

Se a entrada IN for 0, o argumento de tempo decorrido é mantido e a saída Q vai para 0.

Se a entrada IN for 1, o tempo decorrido é zerado e então incrementado até atingir o valor contido no argumento de tempo desejado. Ao atingir este valor, a saída Q vai para 1, ficando nesse estado até a entrada IN ir para 0.

Fluxograma:

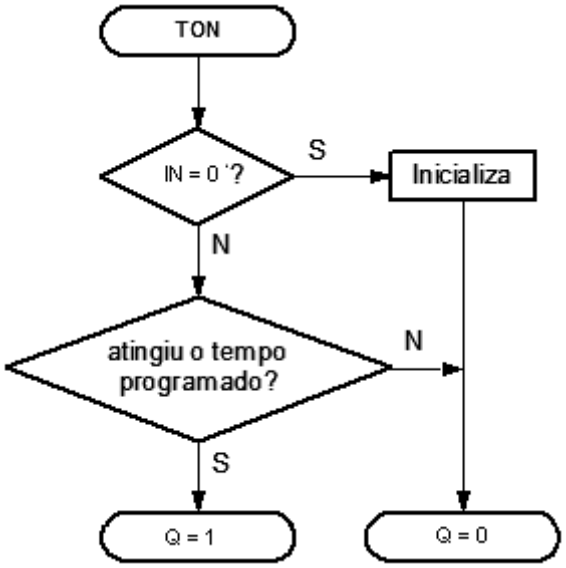
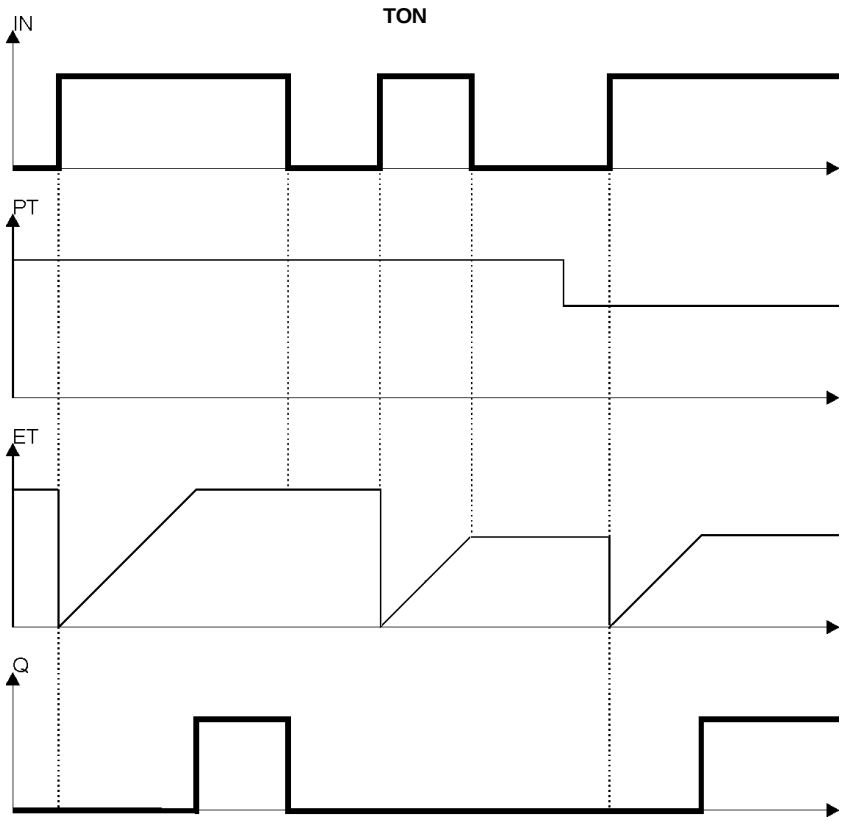
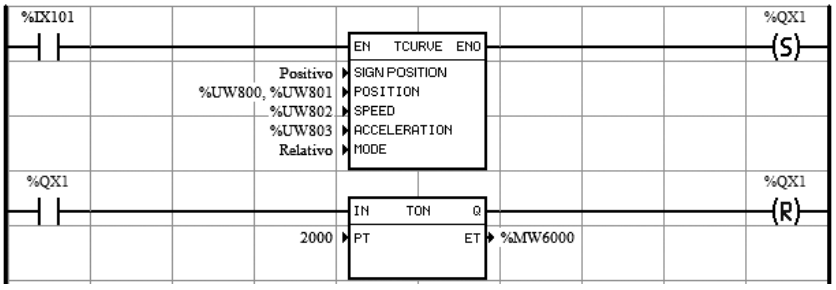


Gráfico:



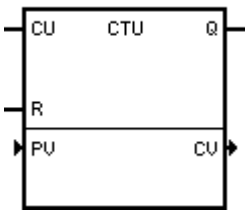
Exemplo:



Quando a entrada digital 1 do drive for 1, um posicionamento baseado nos parâmetros do usuário 800 a 803 é habilitado. Quando este posicionamento termina, a saída digital 1 é setada e o temporizador é habilitado. Após os 2000 ms programados estourarem, a saída digital 1 é resetada.

10.17 Bloco Contador Incremental (CTU)

Figura:



Descrição:

É composto por 1 entrada CU, 1 entrada R, 1 saída Q e 2 argumentos, sendo eles:
- ☑ PV – contagem desejada
- ☑ CV – contagem decorrida

A entrada CU é a entrada de contagem.

A entrada R reseta a contagem.

A saída Q informa se o valor de contagem programado foi atingido.

PV (Contagem Desejada) – CTU:

A contagem desejada é composta por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado da contagem desejada pode ser:
- ☑ constante
- ☑ parâmetro do usuário
- ☑ marcador de word

Para o tipo de dado constante, o valor máximo permitido é 30.000.

CV (Contagem Decorrida) – CTU:

A contagem decorrida é composta por um tipo de dado e um endereço.

O tipo de dado da contagem decorrida pode ser:
- ☑ parâmetro do usuário
- ☑ marcador de word

NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado.

Funcionamento:

Quando a entrada CU for de 0 para 1, o valor de contagem decorrida é incrementado, a menos que a entrada R esteja em 1.

Quando o valor de contagem decorrida atingir o valor de contagem desejado, a saída Q vai para 1, permanecendo nesse estado até que a entrada R vá para 1. Caso contrário, a saída Q é 0.

Enquanto a entrada R estiver em 1, o valor de contagem decorrida é resetado e a contagem não é incrementada.

Fluxograma:

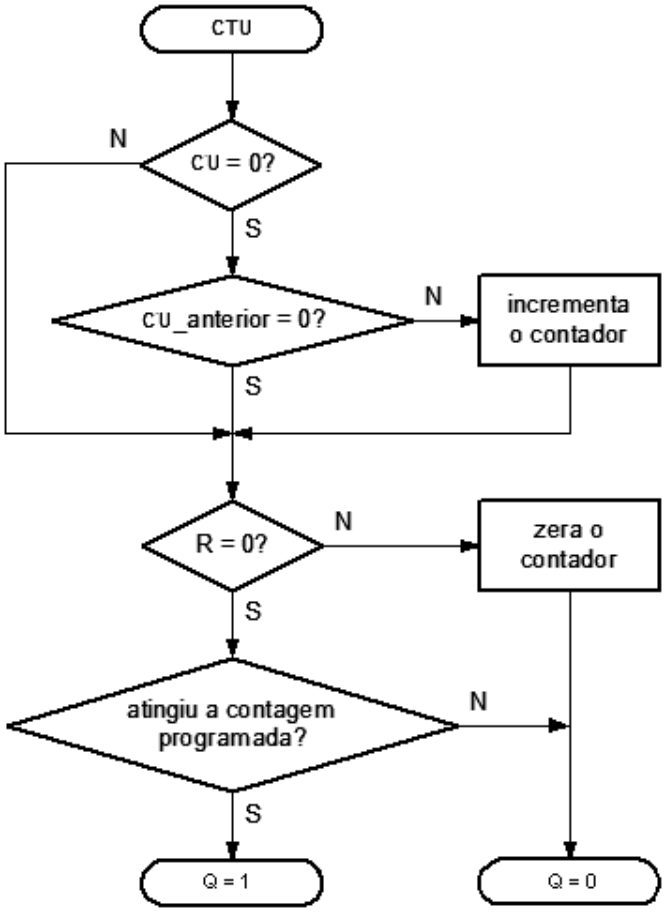
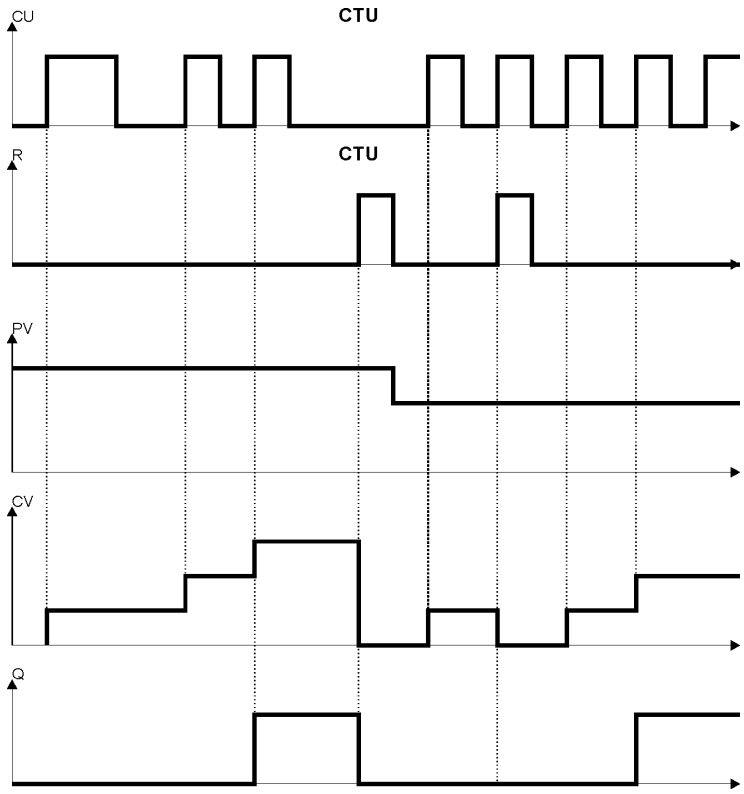
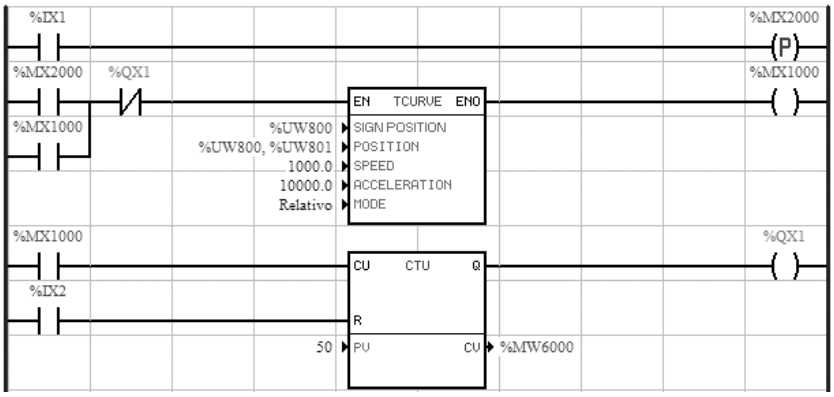


Gráfico:



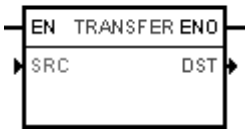
Exemplo:



Se houver uma transição de 0 para 1 na entrada digital 1 ou o marcador de bit 1000 for 1, e a saída digital 1 for 0, um posicionamento TCURVE é habilitado. Na sua conclusão, o marcador 1000 vai para 1, fazendo com que o bloco CTU efetue uma contagem e novamente aciona o posicionamento, desde que a entrada digital 2 seja 0. Quando o contador sentir 50 transições positivas no marcador 1000, ou seja, efetuou 50 posicionamentos, a saída digital 1 vai para 1, fazendo com que um novo posicionamento não seja possível de ser feito, até que a entrada digital 2 for 1, resetando a saída 1.

10.18 Bloco Transfer (TRANSFER)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 2 argumentos, sendo eles:
- ☑ SRC – dado fonte
- ☑ DST – dado destino

A entrada EN é responsável pela habilitação do bloco.

A saída ENO vai para 1 quando o dado destino ter sido atualizado.

SRC (Dado Fonte):

O dado fonte é composto por um tipo de dado e um endereço ou um valor constante, dependendo da escolha do tipo de dado.

O tipo de dado do dado fonte pode ser:
- ☑ constante
- ☑ constante float
- ☑ marcador de bit
- ☑ marcador de word
- ☑ marcador de float
- ☑ marcador de sistema
- ☑ entrada digital
- ☑ saída digital
- ☑ entrada analógica
- ☑ saída analógica
- ☑ parâmetro do usuário
- ☑ parâmetro do sistema
- ☑ parâmetro do drive

DST (Dado Destino):

O dado destino é composto por um tipo de dado e um endereço e é o local onde é salvo o valor do dado fonte.

O tipo de dado do dado destino pode ser:
- ☑ marcador de bit
- ☑ marcador de word
- ☑ marcador de float
- ☑ marcador de sistema
- ☑ saída digital
- ☑ saída analógica
- ☑ parâmetro do usuário
- ☑ parâmetro do sistema
- ☑ parâmetro do drive

NOTA!
Na opção parâmetro do usuário, o valor corrente não é salvo na memória E2PROM, ou seja, este último valor não é recuperado.

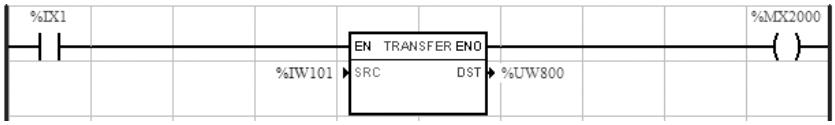
Funcionamento:

A saída ENO vai para 1 se a entrada EN for 1 e após o dado destino ter sido atualizado.

Quando a entrada EN está ativa, o valor contido no dado fonte é transferido para o dado destino. Caso contrário, nada é feito.

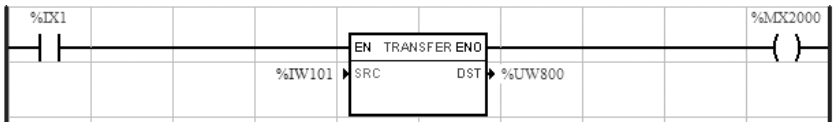
Atenção à compatibilidade quanto aos tipos de dados fonte e destino.

Exemplo:



A entrada digital 1 em 1, habilita o TRANSFER. Com isto o valor contido na entrada analógica 1 pode ser visualizado no parâmetro do usuário 800.

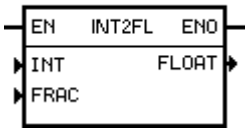
Uma aplicação útil do bloco TRANSFER é a sua utilização para habilitar o motor à partir, por exemplo, de uma entrada digital. Assim, SRC teria uma entrada digital como valor, e DST o marcador de sistema %SW0. Lembrar que o motor só é habilitado se o mesmo já tiver sido habilitado no inversor CFW-09. Isso pode ser programado, por exemplo, na entrada digital 1 do drive.



NOTA!
A leitura ou escrita de parâmetros do drive consome um tempo considerável tornando isso lento quando existirem muitas requisições simultâneas. Isto pode ocasionar falhas na monitoração on-line se a mesma estiver lendo parâmetros do drive.

10.19 Bloco Converte Inteiro para Ponto Flutuante (INT2FL)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 3 argumentos, sendo eles:
- ☑ parte inteira – word (ver item 9.4.7)
- ☑ parte fracionária – word (ver item 9.4.8)
- ☑ float (ver item 9.4.9)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO é uma cópia do valor da entrada EN.

Funcionamento:

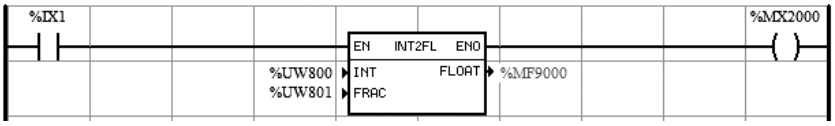
A entrada EN sempre transfere o seu valor para a saída ENO.

Enquanto a entrada EN for 1, os valores contidos na word inteira e word fracionária são transferidos ao marcador de float.

A word inteira e fracionária representam um número no formato 16.16. A word inteira representa um número inteiro e pode variar de -32768 a 32767. A word fracionária representa um número decimal sempre positivo que pode variar de 0.0 a 0.9999847.

Exemplo: A conversão de uma word inteira, igual a 3, e uma word fracionária, igual a 8192, resulta num valor 3.125 em ponto flutuante, pois 8192 / 65536 = 0.125.

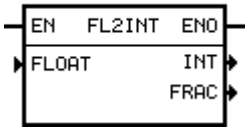
Exemplo:



Converte o valor do parâmetro do usuário 800 e 801 para o marcador de float 9000. Lembrando que o parâmetro 800 representa a parte inteira e o parâmetro 801 a parte fracionária.

10.20 Bloco Converte Ponto Flutuante para Inteiro e Fracionário (FL2INT)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 3 argumentos, sendo eles:
- ☑ float (ver item 9.4.9)
- ☑ parte inteira – word (ver item 9.4.7)
- ☑ parte fracionária – word (ver item 9.4.8)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO é uma cópia do valor da entrada EN.

Funcionamento:

A entrada EN sempre transfere o seu valor para a saída ENO.

Enquanto a entrada EN for 1, o valor contido no float é transferido para a word inteira e para a word fracionária.

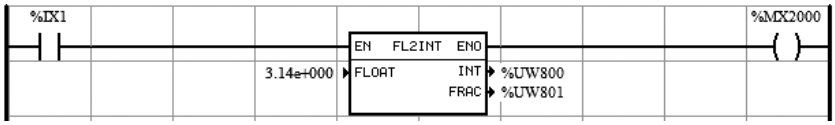
A word inteira e fracionária representam um número no formato 16.16. A word inteira representa um número inteiro e pode variar de -32768 a 32767. A word fracionária representa um número decimal sempre positivo que pode variar de 0.0 a 0.9999847.

Exemplo: A conversão do float -5.5 resulta em uma word inteira, igual a -5 e uma word fracionária, igual a 32768, que representa 32768 / 65536 = 0.5.

Se o valor float for maior que 32767, na conversão seu valor é saturado resultando numa word inteira, igual a 32767 e uma word fracionária = 65535, que representa 65535 / 65536 = 0.9999847.

Se o valor float for menor que -32768, na conversão seu valor é saturado resultando numa word inteira, igual a -32768 e uma word fracionária = 65535, que representa 65535 / 65536 = 0.9999847.

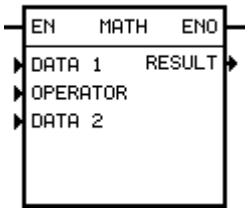
Exemplo:



Quando a entrada digital 1 for 1, o valor de ? é convertido para os parâmetros do usuário 800 e 801, onde o parâmetro 800 terá o valor de 3, e o parâmetro 801 terá o valor de 9175, que representa 0,14.

10.21 Bloco Aritmético (MATH)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 4 argumentos, sendo eles:
- ☑ operador
- ☑ float 1 (ver item 9.4.9)
- ☑ float 2 (ver item 9.4.9)
- ☑ float resultado (ver item 9.4.9)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO é uma cópia do valor da entrada EN.

Como todos os tipos de dado deste bloco são constante float ou marcador de float, é útil fazer o uso dos blocos INT2FL e FL2INT.

Operador:

O operador é sempre constante.

Possui as opções:
- ☑ Adição
- ☑ Subtração
- ☑ Multiplicação
- ☑ Divisão

Funcionamento:

A entrada EN sempre transfere o seu valor para a saída ENO.

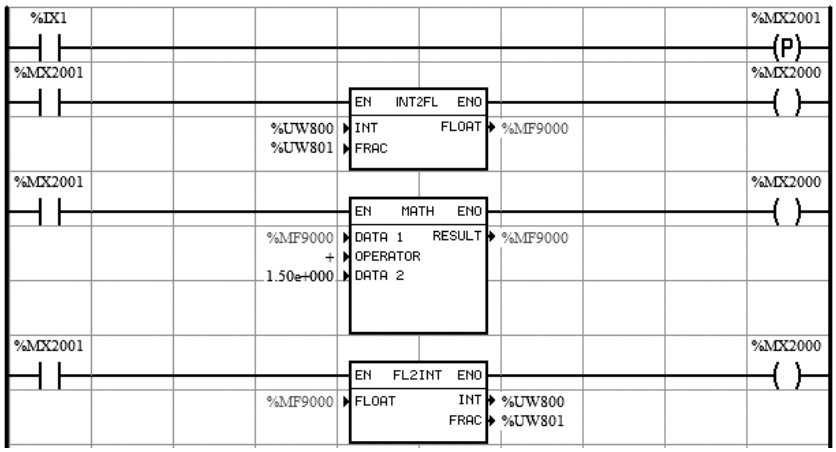
Enquanto a entrada EN for 1, é executada a operação matemática programada entre os argumentos float 1 e float 2, e transferindo o resultado para o float resultado.

A operação executada é dada por:

[float resultado] = [float 1] [operador] [float 2]

Numa divisão pela constante 0, é gerado um “warning” na compilação. Caso a divisão seja efetuada com um parâmetro ou marcador no denominador, essa verificação não acontece, porém, em ambos os casos o valor é saturado aos valores máximo ou mínimo de float, dependendo do valor do numerador ser maior ou menor que 0. Para efeitos do sinal da saturação, zero é considerado com sinal positivo.

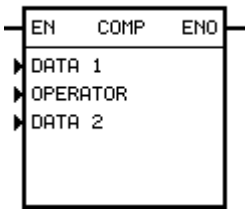
Exemplo:



A cada pulso dado na entrada digital 1, o valor do parâmetro do usuário 800 e 801 é incrementado de 1,5 , lembrando que o valor do parâmetro 800 representa a parte inteira e o parâmetro 801 representa a parte fracionária.

10.23 Bloco Comparador (COMP)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 4 argumentos, sendo eles:
- ☑ operador
- ☑ float 1 (ver item 9.4.9)
- ☑ float 2 (ver item 9.4.9)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO é uma cópia do valor da entrada EN.

Como todos os tipos de dado deste bloco são constante float ou marcador de float, é útil fazer o uso dos blocos INT2FL e FL2INT.

Operador:

O operador é sempre constante.

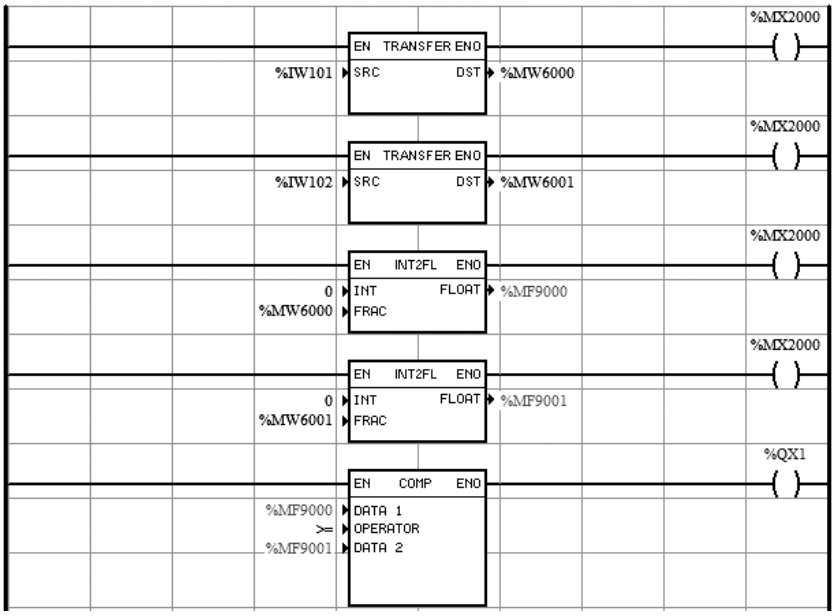
Possui as opções:
- ☑ Igual a
- ☑ Diferente de
- ☑ Maior que
- ☑ Maior ou igual a
- ☑ Menor que
- ☑ Menor ou igual a

Funcionamento:

Quando a entrada EN é 0, o bloco não é executado e a saída ENO vai para 0.

Enquanto a entrada EN for 1 e a comparação [float 1] [operador] [float 2] é verdadeira, a saída ENO vai para 1. Caso contrário, vai para 0.

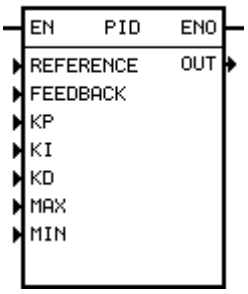
Exemplo:



Neste exemplo, se o valor contido na entrada analógica 1 do drive for maior ou igual ao valor contido na entrada analógica 2 do drive, liga a saída digital 1. Caso contrário, desliga a saída digital 1.

10.24 Bloco PID (PID)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 3 tipos de argumentos, sendo eles:

- ☑ sinais (referência, feedback, saída de controle)
- ☑ ganhos (KP, KI, KD)
- ☑ limites (máximo, mínimo) (ver item 9.4.10)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO é uma cópia do valor da entrada EN.

Como todos os tipos de dado deste bloco são constante float ou marcador de float, é útil fazer o uso dos blocos INT2FL e FL2INT.

Sinais:

Os sinas são compostos por 3 partes:
- ☑ float – referência (ver item 9.3.9)
- ☑ float – realimentação (ver item 9.3.9)
- ☑ float – controle (ver item 9.3.9)

Ganhos:

Os ganhos são compostos por 3 partes:
- ☑ float – ganho proporcional (Kp) (ver item 9.3.9)
- ☑ float – ganho integral (Ki) (ver item 9.3.9)
- ☑ float – ganho derivativo (Kd) (ver item 9.3.9)

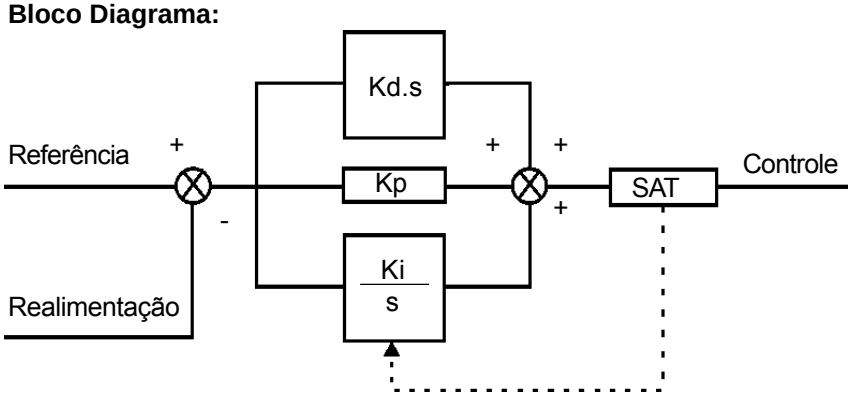
Funcionamento:

A entrada EN sempre transfere o seu valor para a saída ENO.

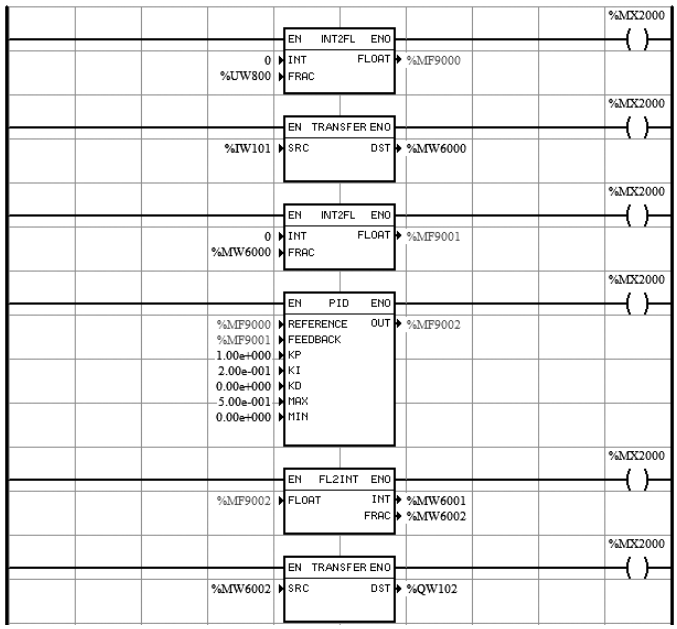
Enquanto a entrada EN for 1, o bloco é executado. Caso contrário, os argumentos são resetados.

ATENÇÃO!
No máximo 2 blocos de PID podem estar ativos por vez. A partir do terceiro, não são executados, mesmo que estejam ativos em sua entrada EN.

Bloco Diagrama:



Exemplo:



Resumidamente, o valor de referência é dado pelo parâmetro do usuário 800, que por sua vez é convertido para o marcador de float 9000. O valor do sinal de realimentação é dado pelo valor contido na entrada analógica 1 do drive, que é transferido ao marcador de word 6000 e convertido ao marcador de float 9001. A saída de controle do bloco PID é o marcador 9002, que é convertido para os marcadores de word 6001 e 6002. O valor contido no marcador de word 6002 é transferido para a saída analógica 2 do drive.

10.25 Bloco Saturação (SAT)

Figura:

Descrição:

É composto por 1 entrada EN, 1 saída ENO e 2 tipos de argumentos, sendo eles:

☑ valores (entrada, saída) (ver item 9.4.11)
☑ limites (máximo, mínimo) (ver item 9.4.10)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO indica quando ocorre uma saturação.

Como todos os tipos de dado deste bloco são constante float ou marcador de float, é útil fazer o uso dos blocos INT2FL e FL2INT.

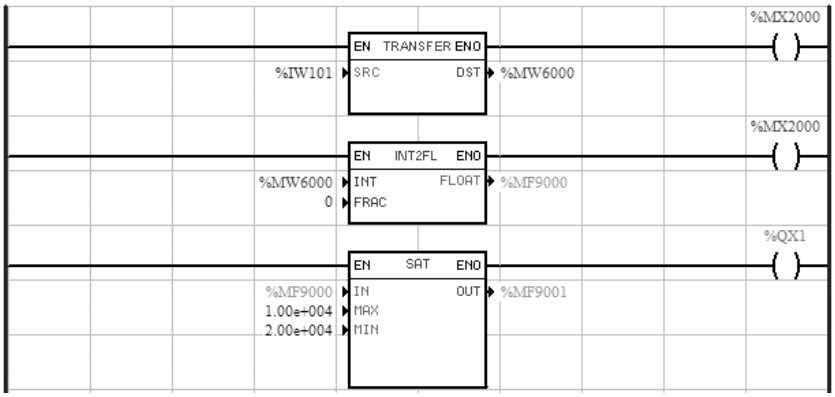
Funcionamento:

Se a entrada EN é 0, o bloco não é executado e a saída ENO vai para 0.

Enquanto a entrada EN for 1, o bloco é executado. A saída ENO só vai para 1 se houver uma saturação. Caso contrário, a saída ENO fica em 0.

A idéia do bloco é transferir os dados da entrada para a saída se estiverem dentro dos limites programados. Se estes valores forem maiores ou menores que os valores máximos e mínimos programados, a valor da saída é saturado com estes valores.

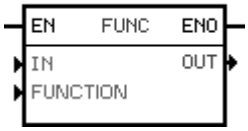
Exemplo:



O valor contido na entrada analógica 1 do drive é transferido para o marcador de word 6000, que por sua vez é convertido para o marcador de float 9000. O valor lido da entrada analógica é um valor entre 0 e 32767. O bloco SAT faz com que no marcador de float 9001 seja lido somente um valor entre 10000 e 20000.

10.26 Bloco Função Matemática (FUNC)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 3 argumentos, sendo eles:
- ☑ função
- ☑ valores (entrada, saída) (ver item 9.4.11)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO é uma cópia do valor da entrada EN.

Como todos os tipos de dado deste bloco são constante float ou marcador de float, é útil fazer o uso dos blocos INT2FL e FL2INT.

Função:

A função é sempre constante.

Possui as opções:
- ☑ absoluto (módulo)
- ☑ negativo
- ☑ raiz quadrada
- ☑ seno
- ☑ coseno
- ☑ tangente
- ☑ arco seno
- ☑ arco coseno
- ☑ arco tangente

Funcionamento:

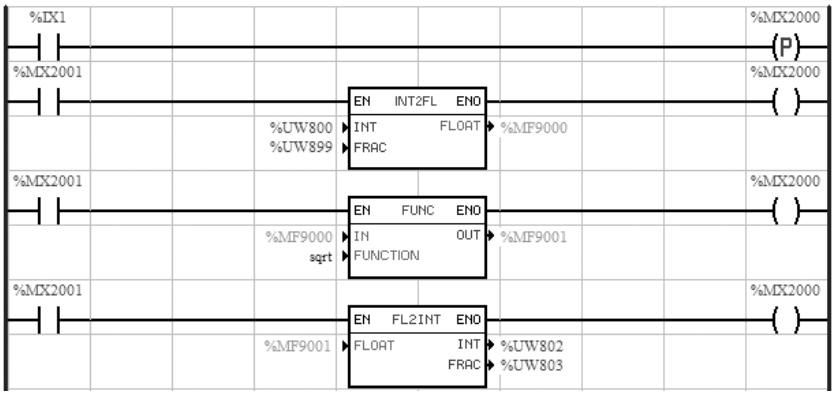
A entrada EN sempre transfere o seu valor para a saída ENO.

Enquanto a entrada EN for 1, o bloco é executado.

As fórmulas são:

- ☑ absoluto: [saída] = | [entrada] |
- ☑ negativo: [saída] = - [entrada]
- ☑ raiz quadrada: [saída] = sqrt([entrada])
- ☑ seno: [saída] = sin([entrada]) → [entrada] em radianos
- ☑ coseno: [saída] = cos([entrada]) → [entrada] em radianos
- ☑ tangente: [saída] = tag([entrada]) → [entrada] em radianos
- ☑ arco seno: [saída] = asin([entrada]) → [saída] em radianos
- ☑ arco coseno: [saída] = acos([entrada]) → [saída] em radianos
- ☑ arco tangente: [saída] = atag([entrada]) → [saída] em radianos

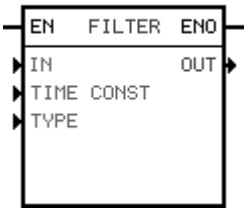
Exemplo:



Na transição de 0 para 1 na entrada digital 1, os parâmetros do usuário 800 e 801 são convertidos para o marcador de float 9000. Então é calculado a raiz quadrada do valor contido no marcador de float 9000 e salvo no marcador de float 9001. O valor do marcador de float 9001 é convertido para os parâmetros do usuário 802 e 803.

10.27 Bloco Filtro 1ª Ordem (FILTER)

Figura:



Descrição:

É composto por 1 entrada EN, 1 saída ENO e 2 argumentos, sendo eles:
- ☑ valores (entrada, saída) (ver item 9.4.11)
- ☑ tipo de filtro
- ☑ float – constante de tempo [segundos] (ver item 9.4.9)

A entrada EN é responsável pela habilitação do bloco.

A saída ENO é uma cópia do valor da entrada EN.

Como todos os tipos de dado deste bloco são constante float ou marcador de float, é útil fazer o uso dos blocos INT2FL e FL2INT.

Tipo:

O tipo de filtro é uma constante, que pode ser:
- ☑ passa baixa
- ☑ passa alta

Funcionamento:

A entrada EN sempre transfere o seu valor para a saída ENO.

Enquanto a entrada EN for 1, o bloco é executado. Caso contrário, os argumentos são resetados.

A fórmula do Filtro é dada por:

para filtros passa baixa:

$$saída = \frac{entrada}{constante\ de\ tempo\ * s + 1}$$

para filtros passa alta:

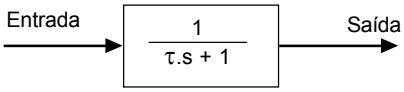
$$saída = \frac{entrada * constante\ de\ tempo * s}{constante\ de\ tempo\ * s + 1}$$

NOTA!
A constante de tempo é dada em segundos.

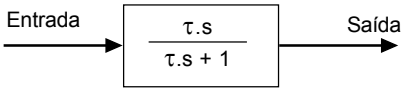
ATENÇÃO!
No máximo 2 blocos de filtro podem estar ativos por vez. A partir do terceiro, não são executados, mesmo que estejam ativos em sua entrada EN.

Bloco Diagrama:

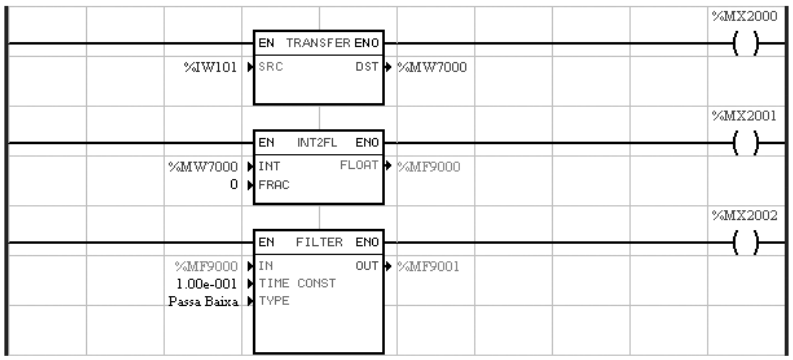
Filtro Passa Baixa



Filtro Passa Alta



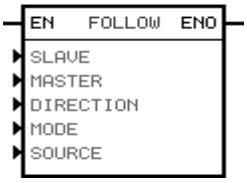
Exemplo:



O valor contido na entrada analógica 1 do drive é transferido ao marcador de word 6000. Este marcador de word 6000 é convertido para o marcador de float 9000. O marcador de float 9000 é a entrada do filtro, cuja constante de tempo é 0,1s, resultando no marcador de float 9001.

10.28 Follow

Figura:



Descrição:

É formado por 1 entrada EN, 1 saída ENO e 2 argumentos sendo:
☑ Direção
☑ Relação de sincronismo

A entrada EN habilita o escravo seguir o mestre baseado nos dados recebidos pela rede CAN.

A saída ENO informa se o escravo atingiu o sincronismo.

Relação de Sincronismo

A relação de sincronismo é formada por 1 tipo de dado e 2 endereços ou constantes, dependendo da escolha do tipo de dado.

O tipo de dado pode ser:
☑ constante
☑ parâmetro do usuário
☑ marcador de word

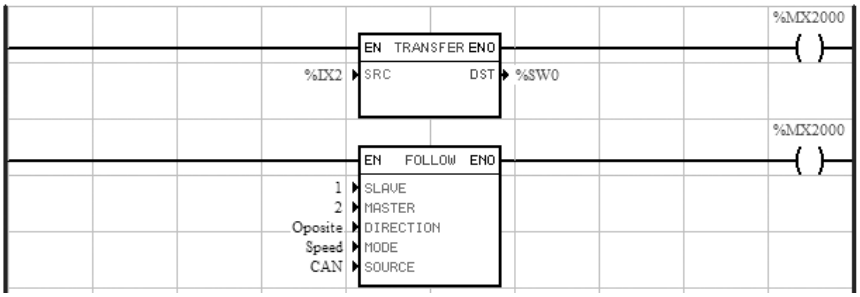
Os endereços ou constantes são destinados a relação do mestre e relação do escravo.

Funcionamento:

Se o drive mestre estiver enviando os dados via rede CAN e a entrada EN do bloco seguidor estiver ativa, o motor escravo segue o motor mestre com os valores da relação de sincronismo em malha de velocidade.

Somente quando o motor escravo atingir a relação especificada do motor mestre, a saída ENO é setada.

Exemplo:



Se o mestre está enviando os dados via rede CAN, o motor escravo roda 1/2 vezes a velocidade do motor mestre.

10.29 CAN2MS

Figura:

Descrição:

É formado por 1 entrada EN e 1 saída ENO.

A entrada EN é responsável por manter o mestre enviar as referências de velocidade e posição via rede CAN ao escravo.

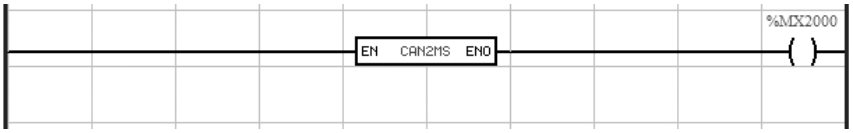
A saída ENO informa se a rede CAN está habilitada.

Funcionamento:

Quando este bloco é habilitado, a PLC1 começa a enviar as referências de velocidade e posição via rede CAN.

NOTA!
Se o bloco não for habilitado no projeto do mestre, o escravo não seguirá o mestre.

Exemplo:



Aqui, a comunicação CAN é habilitada automaticamente e a placa PLC inicia a transmissão da referência de velocidade e posição ao escravo.

MONITORAÇÃO ON-LINE UTILIZANDO WLP

ATENÇÃO!

A monitoração foi introduzida a partir da versão V1.50 do firmware da PLC1 e da V4.00 do software de programação WLP. Para versões inferiores as citadas a monitoração on-line não existe.

11.1 INTRODUÇÃO

A monitoração on-line e feita através da serial RS-232 da placa PLC1 da mesma maneira que o programa Ladder é carregado para a placa PLC1. Ou seja, uma vez o programa ladder compilado e carregado é possível através da serial RS-232 utilizar o programa WLP para representar gráfica e numericamente o estado lógico do programa ladder. Através da monitoração on-line é possível visualizar os estados lógicos de contatos e bobinas do programa ladder bem como o valor numérico atual de marcadores de word, float e parâmetros do drive e da PLC1.

11.2 TOOLBAR E MENU DE MONITORAÇÃO ON-LINE

11.2.1 TOOLBAR MONITORAÇÃO ON-LINE

Nesta toolbar estão todas as funções relativas à monitoração on-line que são :

- MONITORAÇÃO DO LADDER

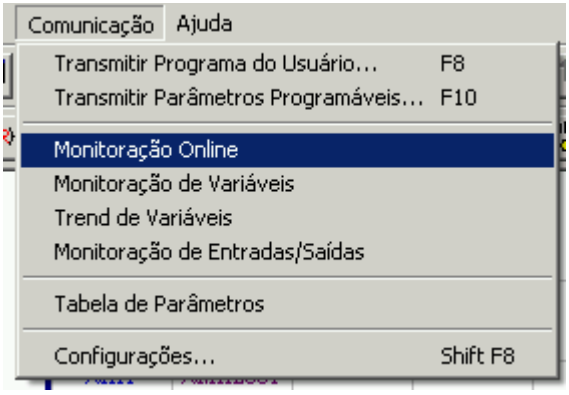
- MONITORAÇÃO DE VARIÁVEIS

- TREND DE VARIÁVEIS

- MONITORAÇÃO DE ENTRADAS E SAÍDAS

11.2.2 MENU DE MONITORAÇÃO ON-LINE

Todas as funções da monitoração on-line podem ser utilizadas individualmente ou em conjunto, ou seja, todas utilizam o mesmo canal de comunicação com a placa PLC1 que é a serial RS-232 de forma compartilhada. Então deve ficar bem claro que quanto mais funções de monitoração estão sendo utilizadas mais informações serão requeridas à placa PLC1 tornando a monitoração mais lenta em função disso.



O menu de monitoração on-line tem as mesmas funções da toolbar mais está sempre disponível no menu comunicação.

11.3 MONITORAÇÃO DO LADDER

Após o programa ladder compilado e carregado na PLC1 é possível monitorar o ladder pressionando o botão de monitoração on-line .
Neste momento o WLP tentará estabelecer a comunicação com a placa PLC1 testando a comunicação serial com a mesma, se a comunicação estiver OK a seguinte mensagem aparecerá no status bar que está na parte inferior do WLP conforme figura a seguir:

Nesta mesma barra existirá um indicador tipo LED da cor azul que ficará piscando indicando que a comunicação está operando .
Casso neste instante ocorra alguma falha de comunicação uma caixa abrirá com a informação da falha e possível solução e a monitoração on-line será desativada.
Uma vez estando à monitoração on-line ativa todas as ferramentas de edição ficarão desativadas e a janela de edição ira mostrar o estado lógico do programa em ladder. Para desativar a monitoração on-line basta pressionar o botão de monitoração on-line novamente.

A seguir será descrito a representação gráfica do estado lógico para contatos e bobinas em monitoração on-line:

CONTATO NORMAL ABERTO CONDUZINDO

CONTATO NORMAL ABERTO NÃO CONDUZINDO

CONTATO NORMAL FECHADO CONDUZINDO

CONTATO NORMAL FECHADO NÃO CONDUZINDO

BOBINA ENERGIZADA

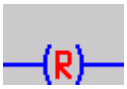
BOBINA DESENERGIZADA

BOBINA NEGADA ENERGIZADA

BOBINA NEGADA DESENERGIZADA

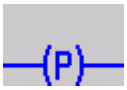
BOBINA SETA ENERGIZADA

BOBINA SETA DESENERGIZADA



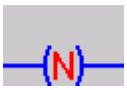
BOBINA RESETA ENERGIZADA

BOBINA RESETA DESENERGIZADA

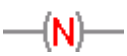


BOBINA TRANSIÇÃO POSITIVA ENERGIZADA

BOBINA TRANSIÇÃO POSITIVA DESENERGIZADA



BOBINA DE TRANSIÇÃO NEGATIVA ENERGIZADA

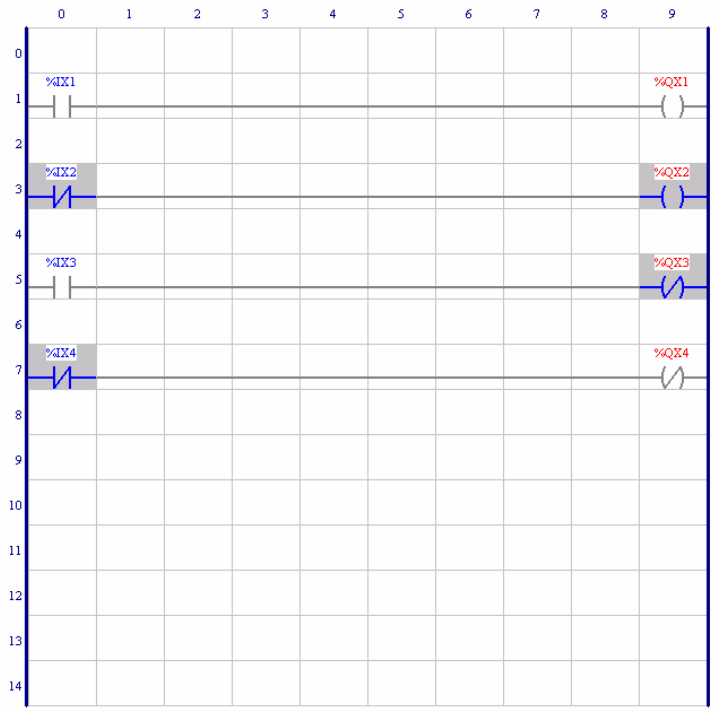


BOBINA DE TRANSIÇÃO NEGATIVA DESENERGIZADA

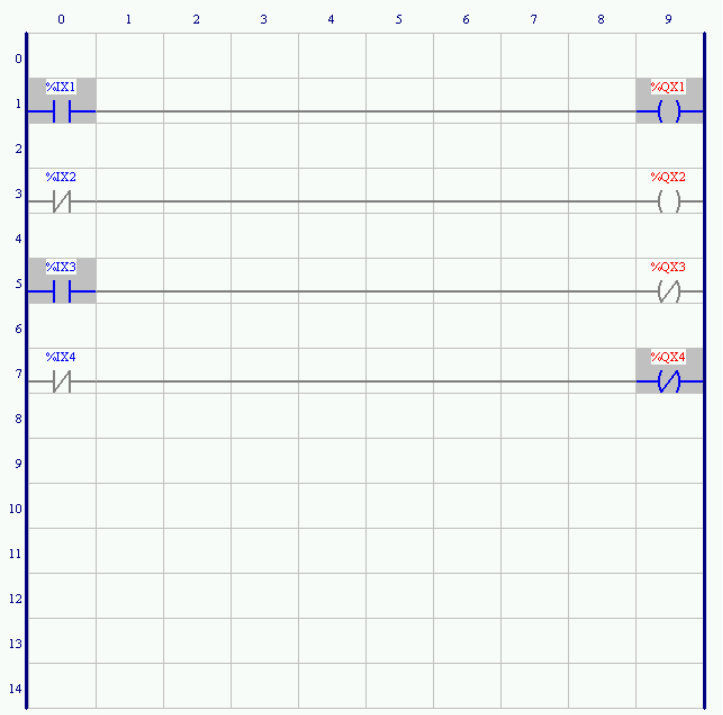
ATENÇÃO!

Deve ficar bem claro que o estado dos contatos refere-se à capacidade de os mesmos estarem ou não conduzindo para a próxima fase do ladder em questão, deve sempre se lembrar a analogia que o ladder se propõe que é a de um circuito elétrico com contatores e seus respectivos contatos. Da mesma maneira o estado das bobinas referem-se as mesmas estarem energizada ou não, quando uma bobina estiver energizada, significa que os seus contatos normais aberto estão conduzindo e seus contatos normais fechados não estão conduzindo.

Para ficar mais claro a representação gráfica veja o exemplo a seguir:



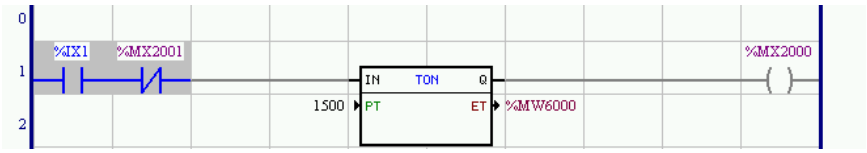
Nesta primeira figura mostro o estado lógico do programa em ladder com as quatro entradas digitais (%IX1...%IX4) sem sinal em seus bornes, ou seja desativadas. Fica evidente que nos contatos normalmente fechados com as entradas desativadas os mesmos estão conduzindo no ladder, da mesma maneira a bobina negada esta energizada não recebendo nenhum sinal do contato %IX3.
Agora analisaremos o mesmo exemplo mais com as quatro entradas recebendo 24Vcc em seus bornes, ou seja ativadas.



Agora os estado lógico dos contatos e bobinas são exatamente o contrário dos anteriores.

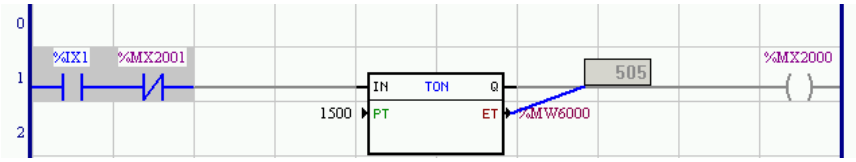
11.3.1 MONITORAÇÃO DE VALORES NÚMERICOS NO LADDER

Quando utilizamos blocos de função no ladder como os blocos SCURVE e TCURVE, utilizamos variáveis numéricas com marcadores de word, marcadores de float e parâmetros do drive ou PLC. A monitoração das mesmas é feita pelo simples clique no conector relacionado a variável numérica conforme figura a seguir:



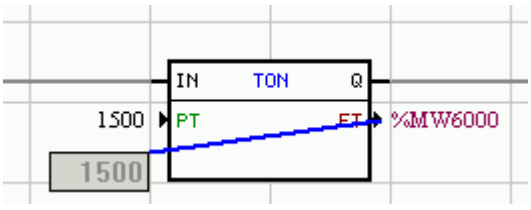
Supondo que você tenha um programa que você deseje monitorar o tempo atual de um temporizador que está no marcador de word %MW6000 conforme figura anterior para tal operação vá com o mouse até a seguinte posição e clique nele:

Após cliquar no ponto em questão a seguinte caixa aparecerá:



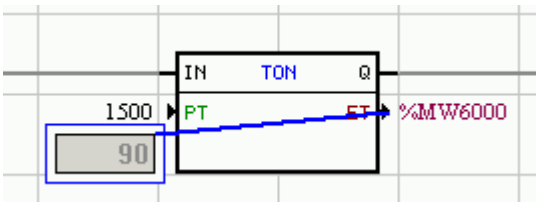
A caixa em questão sempre estará ancorada ao conector cliquado e mostrará o valor da variável relacionada a esse conector, no exemplo mostrado a caixa em questão está monitorando o valor do marcador de word %MW6000.

A caixa de monitoração pode ser posicionada em qualquer local da janela de edição do ladder, para tal basta cliquar na caixa mantendo o botão do mouse pressionado e arrastá-la para a posição desejada, a linha em questão continuará ancorada no conector relativo a variável.



CAIXA REPOSICIONADA

Para apagar a caixa de monitoração basta cliquar na mesma de modo a selecioná-la e após pressionar a tecla DEL.

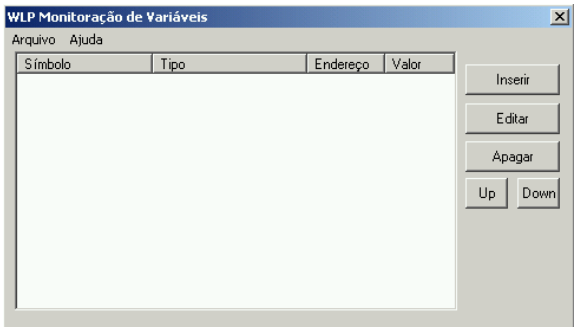


CAIXA SELECIONADA

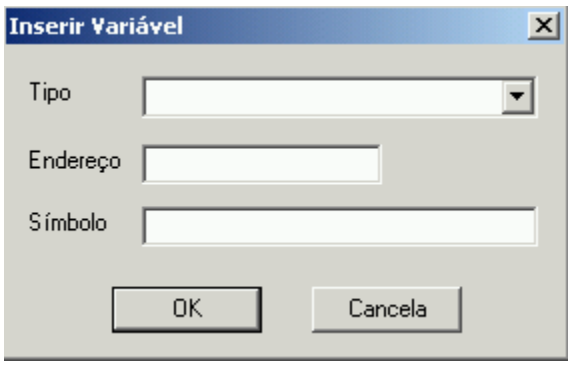
Para remover a seleção basta pressionar a tecla ESC.

11.4 MONITORAÇÃO DE VARIÁVEIS

Através do dialogo de monitoração de variáveis é possível verificar o estado de variáveis utilizadas no programa ladder independente de estar ou não monitorando o ladder. Para carregar esse dialogo basta pressionar o botão de monitoração de variáveis . Da mesma maneira que na monitoração on-line neste momento o WLP tentará estabelecer a comunicação com a placa PLC1 testando a comunicação serial com a mesma e efetuará as mesmas operações anteriormente descritas.
O dialogo de monitoração de variáveis tem o seguinte aspecto :

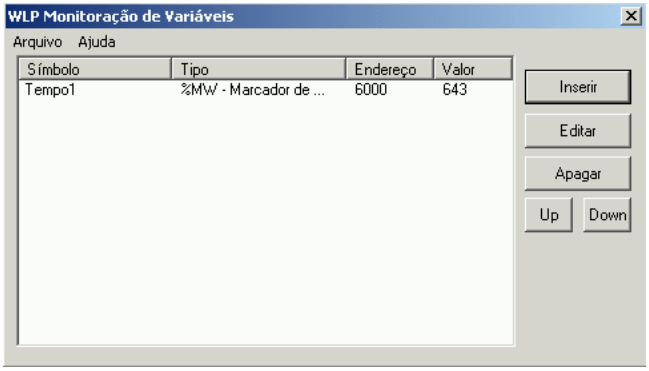


Para inserir novas variáveis basta pressionar o botão inserir que o seguinte dialogo aparecerá:



Neste dialogo basta escolher o tipo, endereço e um símbolo representativo. No exemplo a seguir selecionarei o marcador de word %MW6000:

Ao pressionar o botão OK o dialogo de monitoração de variáveis ficará da seguinte maneira:

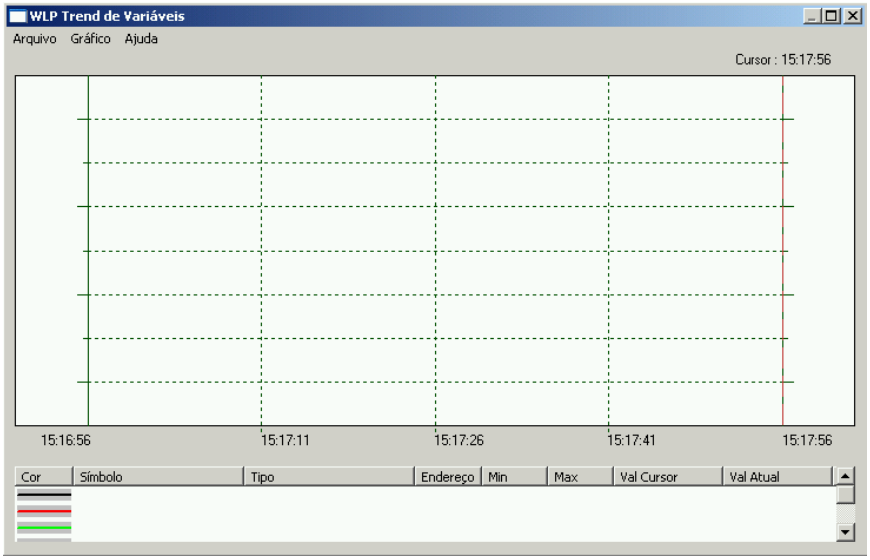


Neste momento o número inserido na coluna denominada valor corresponde ao valor real da variável em questão adquirido da placa PLC1 através da serial RS-232.
Nesta caixa também é possível, editar a variável em questão, apagar a mesma, move-la de posição para cima e para baixo.
Através do menu Arquivo que está no canto superior esquerdo do dialogo é possível salvar e abrir configurações de variáveis feitas do dialogo em questão.

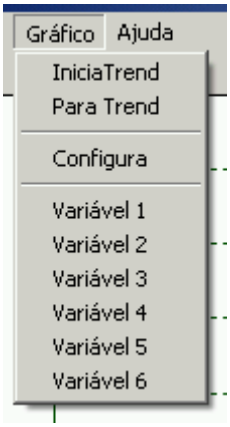
11.5 TREND DE VARIÁVEIS

Através do dialogo de trend de variáveis é possível verificar o estado de variáveis utilizadas no programa ladder independente de estar ou não monitorando o ladder de maneira gráfica semelhante a um plotter de penas. Para carregar esse dialogo basta pressionar o botão de trend de variáveis .

O dialogo de trend de variáveis tem o seguinte aspecto :



Todas as configurações relacionadas ao trend de variáveis estão no menu Gráfico conforme figura a seguir:



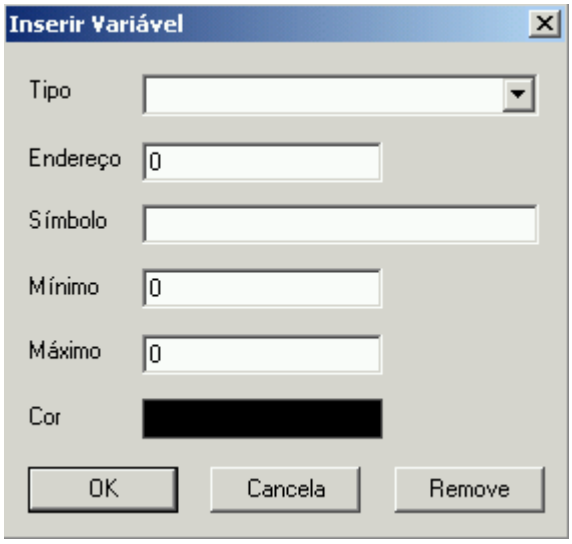
O trend de variáveis tem uma operação um pouco diferente das outras citadas anteriormente, para utilizá-lo e necessário seguir a seqüência citada abaixo :

1º Configurar o gráfico através da opção “Configura” da figura acima.
2º Configura as variáveis a serem plotadas através das opções “Variável 1 a 6” da figura acima.
3º Iniciar o trend através da opção “Inicia Trend” da figura acima.

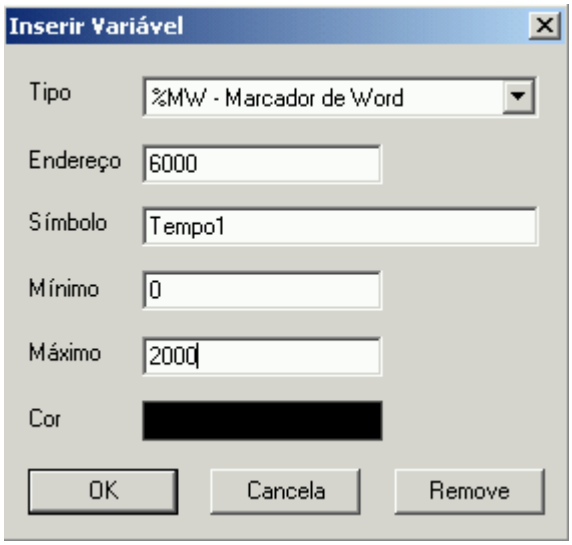
11.5.1 DIALOGO DE CONFIGURAÇÃO DO TREND

Neste dialogo é possível selecionar o ciclo de leitura das variáveis que corresponde a intervalo de tempo entre cada leitura das variáveis selecionadas. Escala do eixo X que corresponde a quantidade de tempo que será possível visualizar no gráfico.

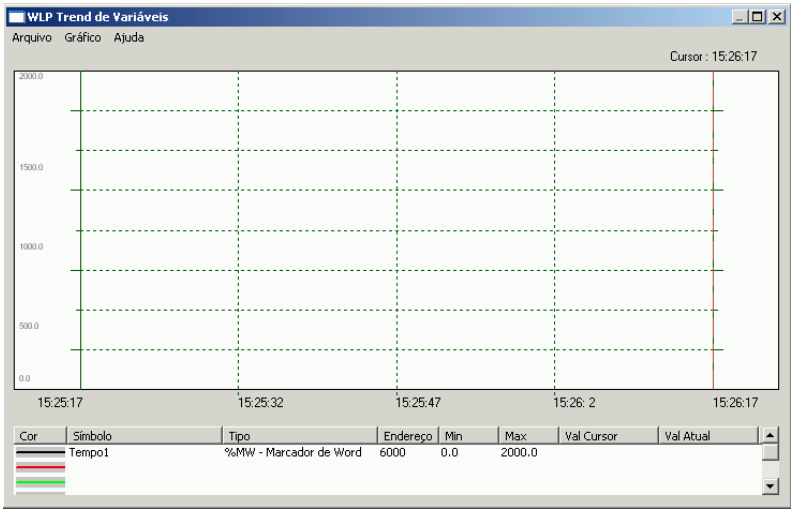
11.5.2 DIALOGO DE CONFIGURAÇÃO DE VARIÁVEIS



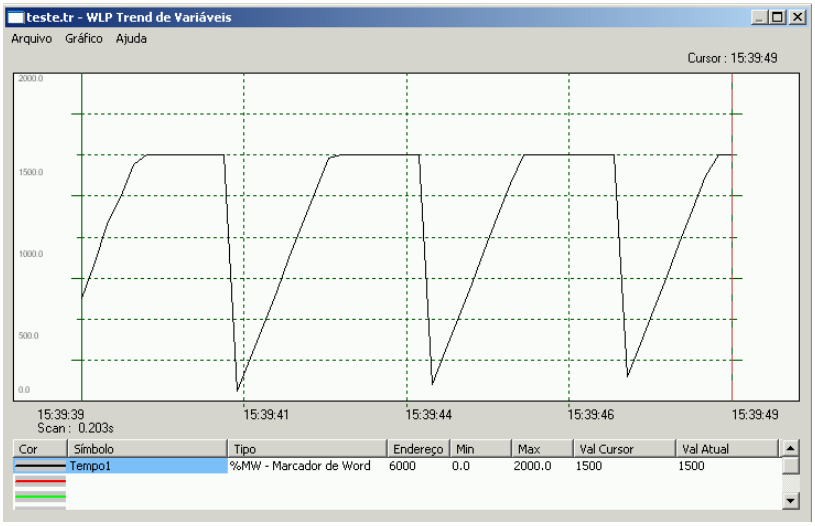
Neste dialogo basta escolher o tipo, endereço, um símbolo representativo, mínimo e máximo e cor da variável. No exemplo a seguir selecionarei o marcador de word %MW6000:



Ao pressionar o botão OK o dialogo de trend de variáveis ficará da seguinte maneira:



A pressionar a opção “Inicia Trend” da mesma maneira que na monitoração on-line neste momento o WLP tentará estabelecer a comunicação com a placa PLC1 testando a comunicação serial com a mesma e efetuará as mesmas operações anteriormente descritas. Uma vez estabelecida a comunicação serial o trend irá adquirir as variáveis conforme o ciclo solicitado e desenhará as mesmas na tela conforme figura a seguir :

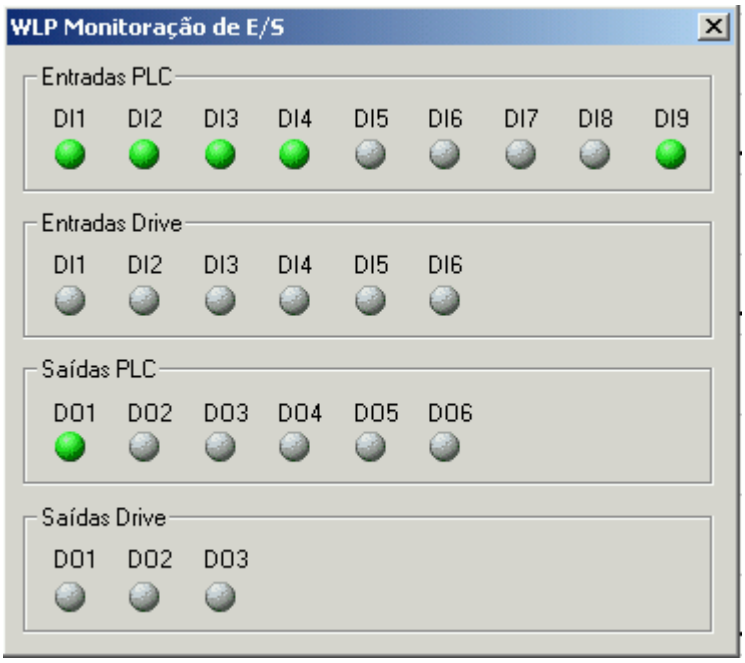


Nesta caixa também é possível, editar a variável em questão, apagar a mesma.
Através do menu Arquivo que está no canto superior esquerdo do dialogo é possível salvar e abrir configurações de trend feitas do dialogo em questão e imprimir o trend em questão.

11.6 MONITORAÇÃO DE ENTRADAS/SAÍDAS

Através do dialogo de monitoração de entradas/saídas e possível verificar o estado das entradas e saídas digitais da placa PLC e do drive. Para carregar esse dialogo basta pressionar o botão de monitoração de entradas/saídas . Da mesma maneira que na monitoração on-line neste momento o WLP tentará estabelecer a comunicação com a placa PLC1 testando a comunicação serial com a mesma e efetuará as mesmas operações anteriormente descritas.

O dialogo de monitoração de entradas/saídas tem o seguinte aspecto :


WLP Monitoração de E/S
Entradas PLC
DI1 DI2 DI3 DI4 DI5 DI6 DI7 DI8 DI9
Entradas Drive
DI1 DI2 DI3 DI4 DI5 DI6
Saídas PLC
DO1 DO2 DO3 DO4 DO5 DO6
Saídas Drive
DO1 DO2 DO3

As entradas/saídas ativas aparecem em verde, as inativas em cinza.

PROTOCOLO MODBUS-RTU NA PLC1

A seguir é fornecida uma explicação sobre o funcionamento da placa PLC1 na rede Modbus-RTU.

O baud rate é definido no parâmetro 765, sendo possíveis os seguintes valores:

1 – 1200bps
2 – 2400bps
3 – 4800bps
4 – 9600bps (ajuste de fábrica)
5 – 19200bps

A comunicação é RS-232C, sem paridade, 8 bits e 2 stop bits.

Para implementarmos uma rede, devemos utilizar os conversores MIW-02, que convertem a RS-232C (ponto a ponto) em RS-485 (multiponto).

O endereço da PLC na rede é definido no parâmetro 764, podendo variar entre 1 e 247 (0 é o endereço para broadcast), tendo como ajuste de fábrica o valor 1.

Obs.: após alterar os parâmetros 764 e/ou 765 o sistema deve ser reinicializado.

O que é possível fazer na PLC1 utilizando o protocolo Modbus-RTU:

1 – Escrita / leitura em parâmetros e marcadores (comandos 3, 6 e 16):
Através do protocolo Modbus-RTU da PLC pode-se ler e escrever em parâmetros da placa (P750...P899), parâmetros do inversor (P000...P413) além de marcadores word e marcadores float. Essa operação pode ser em um único parâmetro ou em um grupo de parâmetros.

2 – Leitura de entradas analógicas (comando 4):
Pode-se ler o valor das entradas analógicas 1 e 2 do drive (endereços 101 e 102). A informação do valor é dada em 15 bits, ou seja para o valor das entradas variando de 0 a 100%, temos um valor lido que vai de 0 a 32767.

3 – Escrita / leitura das entradas e saídas digitais e marcadores de bit (comandos 1, 2, 5 e 15):
Pode-se ler e escrever nas saídas digitais, bem como ler as entradas digitais da PLC ou do inversor, além de leitura e escrita nos marcadores do tipo bit, retentivos ou não. Também tem-se acesso ao marcador de sistema SW0, responsável pela habilitação do inversor (quando escrevendo) ou indicar se o drive está ou não habilitado (quando lendo). Essa operação pode ser em um único bit ou em um grupo.

Obs.: saídas digitais ou marcadores de qualquer tipo, utilizados no programa do usuário, terão prioridade sobre a escrita em relação ao Modbus, ou seja, o programa do usuário sobrescreve o estado imposto pelo protocolo Modbus.

4 – Leitura da identificação da placa (comando 43):
Através do comando 43 pode-se ler dados de identificação da placa tais como, fabricante (WEG), modelo (PLC1.01, por exemplo) e a versão do firmware (V1.50, por exemplo).

Descrição detalhada do protocolo:

12.1 MODBUS-RTU

12.1.1 Introdução ao Protocolo Modbus-RTU

O protocolo Modbus foi inicialmente desenvolvido em 1979. Atualmente, é um protocolo aberto amplamente difundido, utilizado por vários fabricantes em diversos equipamentos. A comunicação Modbus-RTU da placa PLC1 foi desenvolvida baseada em dois documentos:

1. MODBUS Protocol Reference Guide Rev. J, MODICON, June 1996.
2. MODBUS Application Protocol Specification, MODBUS.ORG, may 8th 2002.

Nestes documentos estão definidos o formato das mensagens utilizado pelos os elementos que fazem parte da rede Modbus, os serviços (ou funções) que podem ser disponibilizados via rede, e também como estes elementos trocam dados na rede.

12.1.1.1 Modos de Transmissão

Na especificação do protocolo estão definidos dois modos de transmissão: ASCII e RTU. Os modos definem a forma como são transmitidos os bytes da mensagem. Não é possível utilizar os dois modos de transmissão na mesma rede.

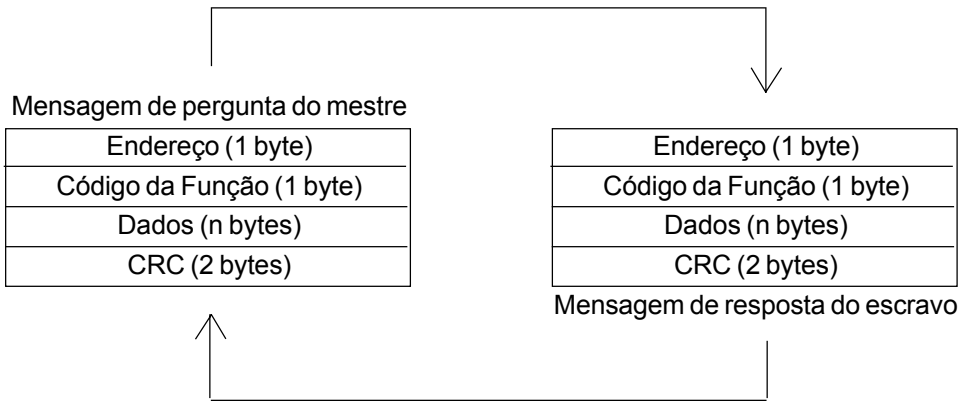
No modo RTU, cada palavra transmitida possui 1 start bit, oito bits de dados, 1 bit de paridade (opcional) e 1 stop bit (2 stop bits caso não se use bit de paridade). Desta forma, a seqüência de bits para transmissão de um byte é a seguinte:

Start	B0	B1	B2	B3	B4	B5	B6	B7	Stop	Stop

No modo RTU, cada byte de dados é transmitido como sendo uma única palavra com seu valor diretamente em hexadecimal. A PLC utiliza somente este modo de transmissão para comunicação, não possuindo portanto, comunicação no modo ASCII.

12.1.1.2 Estrutura das Mensagens no Modo RTU

A rede Modbus-RTU opera no sistema Mestre-Escravo, onde pode haver até 247 escravos, mas somente um mestre. Toda comunicação inicia com o mestre fazendo uma solicitação a um escravo, e este responde ao mestre o que foi solicitado. Em ambos os telegramas (pergunta e resposta), a estrutura utilizada é a mesma: Endereço, Código da Função, Dados e CRC. Apenas o campo de dados poderá ter tamanho variável, dependendo do que está sendo solicitado.



12.1.1.2.1 Endereço

O mestre inicia a comunicação enviando um byte com o endereço do escravo para o qual se destina a mensagem. Ao enviar a resposta, o escravo também inicia o telegrama com o seu próprio endereço. O mestre também pode enviar uma mensagem destinada ao endereço 0 (zero), o que significa que a mensagem é destinada a todos os escravos da rede (broadcast). Neste caso, nenhum escravo irá responder ao mestre.

12.1.1.2.2 Código da Função

Este campo também contém um único byte, onde o mestre especifica o tipo de serviço ou função solicitada ao escravo (leitura, escrita, etc.). De acordo com o protocolo, cada função é utilizada para acessar um tipo específico de dado.

12.1.1.2.3 Campo de Dados

Campo com tamanho variável. O formato e conteúdo deste campo dependem da função utilizada e dos valores transmitidos. Este campo está descrito juntamente com a descrição das funções.

12.1.1.2.4 CRC

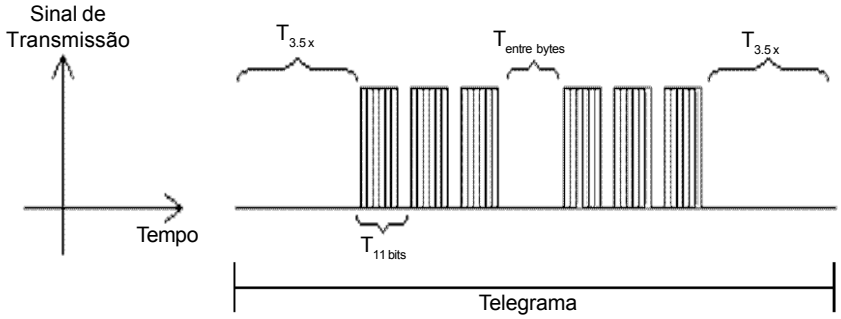
A última parte do telegrama é o campo para checagem de erros de transmissão. O método utilizado é o CRC-16 (Cycling Redundancy Check). Este campo é formado por dois bytes, onde primeiro é transmitido o byte menos significativo (CRC-), e depois o mais significativo (CRC+).
O cálculo do CRC é iniciado primeiramente carregando-se uma variável de 16 bits (referenciado a partir de agora como variável CRC) com o valor FFFFh. Depois executa-se os passos de acordo com a seguinte rotina:

1. Submete-se o primeiro byte da mensagem (somente os bits de dados - start bit , paridade e stop bit não são utilizados) a uma lógica XOR (OU exclusivo) com os 8 bits menos significativos da variável CRC, retornando o resultado na própria variável CRC.
2. Então, a variável CRC é deslocada uma posição à direita, em direção ao bit menos significativo, e a posição do bit mais significativo é preenchida com 0 (zero).
3. Após este deslocamento, o bit de *flag* (bit que foi deslocado para fora da variável CRC) é analisado, ocorrendo o seguinte:
   - ☑ Se o valor do bit for 0 (zero), nada é feito
   - ☑ Se o valor do bit for 1, o conteúdo da variável CRC é submetido a uma lógica XOR com uma valor constante de A001h e o resultado é retornado à variável CRC.
4. Repete-se os passos 2 e 3 até que oito deslocamentos tenham sido feitos.
5. Repete-se os passos de 1 a 4, utilizando o próximo byte da mensagem, até que toda a mensagem tenha sido processada.

O conteúdo final da variável CRC é o valor do campo CRC que é transmitido no final do telegrama. A parte menos significativa é transmitida primeiro (CRC-) e em seguida a parte mais significativa (CRC+).

12.1.1.2.5 Tempo entre Mensagens

No modo RTU não existe um caracter específico que indique o início ou o fim de um telegrama. Desta forma, o que indica quando uma nova mensagem começa ou quando ela termina é a ausência de transmissão de dados na rede, por um tempo mínimo de 3,5 vezes o tempo de transmissão de uma palavra de dados (11 bits). Sendo assim, caso um telegrama tenha iniciado após a decorrência deste tempo mínimo sem transmissão, os elementos da rede irão assumir que o caracter recebido representa o início de um novo telegrama. E da mesma forma, os elementos da rede irão assumir que o telegrama chegou ao fim após decorrer este tempo novamente.
Se durante a transmissão de um telegrama, o tempo entre os bytes for maior que este tempo mínimo, o telegrama será considerado inválido, pois a PLC irá descartar os bytes já recebidos e montará um novo telegrama com os bytes que estiverem sendo transmitidos.
A tabela a seguir nos mostra os tempos para três taxas de comunicação diferentes.



Taxa de Comunicação	$T_{11\ bits}$	$T_{3.5x}$
9600 bits/seg	1.146 ms	4.010 ms
19200 bits/seg	573 μs	2.005 ms

$T_{11\ bits}$ = Tempo para transmitir uma palavra do telegrama.
$T_{entre\ bytes}$ = Tempo entre bytes (não pode ser maior que $T_{3.5x}$).
$T_{3.5x}$ = Intervalo mínimo para indicar começo e fim de telegrama (3.5 x T_{11bits}).

12.1.2 Operação da PLC na Rede Modbus-RTU

As placas PLC operam como escravas da rede Modbus-RTU, sendo que toda a comunicação inicia com o mestre da rede solicitando algum serviço para um endereço na rede. Se a PLC estiver configurada para o endereço correspondente, ela então trata o pedido e responde ao mestre o que foi solicitado.

12.1.2.1 Descrição das Interfaces

As placas PLC utilizam uma interface serial para se comunicar com a rede Modbus-RTU. Existem duas possibilidades para a conexão física entre o mestre da rede e uma PLC:

12.1.2.1.1 RS-232

- ☑ Utilizada para conexão ponto a ponto (entre um único escravo e o mestre).
- ☑ Distância máxima: 10 metros.
- ☑ Níveis de sinal seguem a EIA STANDARD RS-232C.
- ☑ Três fios: transmissão (TX), recepção (RX) e retorno (0V).
- ☑ Deve-se utilizar o módulo RS-232 Serial Interface.

12.1.2.1.2 RS-485

- ☑ Disponível através do conversor MIW-02 conectado à RS-232 da PLC.
- ☑ Utilizada para conexão multiponto (vários escravos e um mestre).
- ☑ Distância máxima: 1000 metros (utiliza cabo com blindagem).
- ☑ Níveis de sinal seguem a EIA STANDARD RS-485.

12.1.2.1.3 Configurações da PLC na Rede Modbus-RTU

Para que a PLC possa se comunicar corretamente na rede, além da conexão física, é necessário configurar o endereço da mesma na rede, bem como a taxa de transmissão.

12.1.2.1.4 Endereço da PLC na Rede

- ☑ Definido através do parâmetro 764.
- ☑ Cada escravo na rede deve possuir um endereço diferente dos demais.
- ☑ O mestre da rede não possui endereço.
- ☑ É necessário conhecer o endereço do escravo mesmo que a conexão seja ponto a ponto.

12.1.2.1.5 Taxa de Transmissão

- ☑ Definida através do parâmetro 765.
- ☑ Taxa de transmissão: 1200, 2400, 4800, 9600 ou 19200 kbits/seg.
- ☑ Paridade: Nenhuma.
- ☑ Todos os escravos, e também o mestre da rede, devem estar utilizando a mesma taxa de comunicação e mesma paridade.

12.1.2.2 Acesso aos Dados da PLC e do Inversor

Através da rede, é possível acessar os parâmetros e as entradas e saídas digitais da PLC e do inversor, bem como marcadores da PLC.

12.1.2.2.1 Funções Disponíveis e Tempos de Resposta

Na PLC, os parâmetros e marcadores foram definidos como sendo registradores do tipo *holding*. Além destes registradores, também é possível acessar diretamente entradas e saídas digitais da placa ou do drive, e também os marcadores de bit, que são acessados utilizando as funções do tipo bit, do Modbus. Para acessar estes bits e registradores, foram disponibilizados os seguintes serviços (ou funções):

☑Read Coils
Descrição: Leitura de bloco de bits internos ou bobinas.
Função: lê marcadores bit e saídas digitais da PLC e do drive.
Código da função: 01.
Broadcast: não suportado.
Tempo de resposta: 5 a 10 ms.

☑Read Input Status
Descrição: Leitura de entradas digitais físicas.
Função: lê entradas digitais da PLC e do drive.
Código da função: 02.
Broadcast: não suportado.
Tempo de resposta: 5 a 10 ms.

☑ Read Holding Registers
Descrição: Leitura de bloco de registradores do tipo *holding*.
Função: lê marcadores word e float, parâmetros da PLC e do inversor.
Código da função: 03.
Broadcast: não suportado.
Tempo de resposta: 5 a 10 ms.

☑ Read Input Registers
Descrição: Leitura de bloco de registradores do tipo *input*.
Função: lê entradas analógicas do inversor.
Código da função: 04.
Broadcast: não suportado.
Tempo de resposta: 5 a 10 ms.

☑ Write Single Coil
Descrição: Escrita em um único bit interno ou bobina.
Função: escreve em marcadores bit e saídas digitais da PLC ou do drive.
Código da função: 05.
Broadcast: suportado.
Tempo de resposta: 5 a 10 ms.

☑ Write Single Register
Descrição: Escrita em um único registrador do tipo *holding*.
Função: escreve em um marcador word, parâmetro da PLC e do inversor.
Código da função: 06.
Broadcast: suportado.
Tempo de resposta: 5 a 10 ms.

☑ Write Multiple Coils
Descrição: Escrita em bloco de bits internos ou bobinas.
Função: escreve em múltiplos marcadores bit ou saídas digitais da PLC e do drive.
Código da função: 15.
Broadcast: suportado.
Tempo de resposta: 5 a 10 ms.

☑ Write Multiple Registers
Descrição: Escrita em bloco de registradores do tipo *holding*.
Função: escreve em múltiplos marcadores word ou float e parâmetros da PLC e do inversor.
Código da função: 16.
Broadcast: suportado.
Tempo de resposta: 10 a 20 ms para cada registrador escrito.

☑ Read Device Identification
Descrição: Identificação do dispositivo.
Função: lê fabricante, modelo e versão do firmware da PLC.
Código da função: 43.
Broadcast: não suportado.
Tempo de resposta: 5 a 10 ms.

Obs.: Os escravos da rede Modbus-RTU são endereçados de 1 a 247. O endereço 0 (zero) é utilizado pelo mestre para enviar uma mensagem comum para todos os escravos (broadcast).

12.1.2.2.2 Endereçamento dos Dados

O endereçamento dos dados na PLC é feito com offset igual a zero, o que significa que o número do endereço equivale ao número dado. Os parâmetros, marcadores, bem como as entradas e saídas digitais, são disponibilizados a partir do endereço 0 (zero).

PARÂMETROS DO DRIVE		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
P000	0	0000h
P100	100	0064h
⋮	⋮	⋮
P413	413	019Dh

PARÂMETROS DA PLC		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
P750	750	02Eeh
P800	800	0320h
⋮	⋮	⋮
P899	899	0383h

MARCADORES WORD RETENTIVOS		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
MW6000	6000	1770h
⋮	⋮	⋮
MW6149	6149	1805h

MARCADORES WORD VOLÁTEIS		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
MW7000	7000	1B58h
⋮	⋮	⋮
MW7799	7799	1E77h

MARCADORES FLOAT		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
MF9000	9000	2328h
⋮	⋮	⋮
MF9099	9099	238Bh

MARCADORES BIT RETENTIVOS		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
MX1000	1000	03E8h
⋮	⋮	⋮
MX1671	1671	0687h

MARCADORES BIT VOLÁTEIS		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
MX2000	2000	07D0h
⋮	⋮	⋮
MX3407	3407	0D4Fh

MARCADOR BIT DE SISTEMA		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
SX0	200	00C8h

ENTRADAS DIGITAIS DA PLC		
Número da Entrada Digital	Endereço Modbus	
	Decimal	Hexadecimal
IX1	1	1h
⋮	⋮	⋮
IX9	9	9h

ENTRADAS DIGITAIS DO DRIVE		
Número da Entrada Digital	Endereço Modbus	
	Decimal	Hexadecimal
IX101	101	0065h
⋮	⋮	⋮
IX106	106	006Ah

SAÍDAS DIGITAIS DA PLC		
Número do bit	Endereço Modbus	
	Decimal	Hexadecimal
QX1	1	1h
⋮	⋮	⋮
QX6	6	6h

SAÍDAS DIGITAIS DO DRIVE		
Número do bit	Endereço Modbus	
	Decimal	Hexadecimal
QX101	101	0065h
QX102	102	0066h
QX103	103	0067h

12.1.3 Descrição Detalhada das Funções

Neste item é feita uma descrição detalhada das funções disponíveis na PLC para comunicação Modbus-RTU. Para a elaboração dos telegramas, é importante observar o seguinte:

- ☑ Os valores são sempre transmitidos em hexadecimal.
- ☑ O endereço de um dado, o número de dados e o valor dos registradores são sempre representados em 16 bits. Por isso, é necessário transmitir estes campos utilizando dois bytes (high e low). Para acessar bits, a forma para representar um bit depende da função utilizada.
- ☑ Os telegramas, tanto para pergunta quanto para resposta, não podem ultrapassar 128 bytes.

12.1.3.1 Função 01 - Read Coils

Lê o conteúdo de um grupo de bits (marcadores de bit, marcador de sistema ou saídas digitais da PLC ou do drive) que necessariamente devem estar em seqüência numérica. Esta função possui a seguinte estrutura para os telegramas de leitura e resposta (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do bit inicial (byte high)	Campo Byte Count (no. De bytes de dados)
Endereço do bit inicial (byte low)	Byte 1
Número de bits (byte high)	Byte 2
Número de bits (byte low)	Byte 3
CRC-	etc...
CRC+	CRC-
	CRC+

Cada bit da resposta é colocado em uma posição dos bytes de dados enviados pelo escravo. O primeiro byte, nos bits de 0 a 7, recebe os 8 primeiros bits a partir do endereço inicial indicado pelo mestre. Os demais bytes (caso o número de bits de leitura seja maior que 8), continuam a seqüência. Caso o número de bits lidos não seja múltiplo de 8,

os bits restantes do último byte devem ser preenchidos com 0 (zero).
Exemplo: leitura das saídas digitais, DO1 a DO6 no endereço 1:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	01h	Endereço do escravo	01h
Função	01h	Função	01h
Bit inicial (high)	00h	Byte Count	01h
Bit inicial (low)	01h	Estado das saídas 1 a 6	35h
No. De bits (high)	00h	CRC-	91h
No. De bits (low)	06h	CRC+	9Fh
CRC-	Edh		
CRC+	C8h		

Observação: não esquecer que as saídas digitais do drive (RL1, RL2 e RL3) são representadas na PLC como DO101, DO102 e DO103, respectivamente.

12.1.3.2 Função 02 - Read Inputs Status

Lê o conteúdo de um grupo de entradas digitais da PLC e do drive, que necessariamente devem estar em seqüência numérica. Esta função possui a seguinte estrutura para os telegramas de leitura e resposta (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do bit inicial (byte high)	Campo Byte Count (no. De bytes de dados)
Endereço do bit inicial (byte low)	Byte 1
Número de bits (byte high)	Byte 2
Número de bits (byte low)	Byte 3
CRC-	etc...
CRC+	CRC-
	CRC+

Cada bit da resposta é colocado em uma posição dos bytes de dados enviados pelo escravo. O primeiro byte, nos bits de 0 a 7, recebe os 8 primeiros bits a partir do endereço inicial indicado pelo mestre. Os demais bytes (caso o número de bits de leitura for maior que 8), continuam a seqüência. Caso o número de bits lidos não seja múltiplo de 8, os bits restantes do último byte devem ser preenchidos com 0 (zero).
☑ Exemplo: leitura das entradas digitais, DI2 a DI7 no endereço 1:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	01h	Endereço do escravo	01h
Função	02h	Função	02h
Bit inicial (high)	00h	Byte Count	01h
Bit inicial (low)	02h	Estado das entradas 2 a 7	21h
No. De bits (high)	00h	CRC-	61h
No. De bits (low)	06h	CRC+	90h
CRC-	59h		
CRC+	C8h		

No exemplo, como o número de bits lidos é menor que 8, o escravo precisou de apenas 1 byte para a resposta. O valor do byte foi 21h, que em binário tem a forma 0010 0001. Como o número de bits lidos é igual a 6, somente nos interessa os seis bits menos significativos, que possuem os valores das entradas digitais de 2 a 7. Os demais bits, como não foram solicitados, são preenchidos com 0 (zero).

Observação: não esquecer que as entradas digitais do drive (DI1...DI6) são representadas na PLC como DI101...DI106, respectivamente.

12.1.3.3 Função 03 - Read Holding Register

Lê o conteúdo de um grupo de marcadores word e float ou parâmetros da PLC ou do drive, que necessariamente devem estar em seqüência numérica. Esta função possui a seguinte estrutura para os telegramas de leitura e resposta (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do registrador inicial (byte high)	Campo Byte Count
Endereço do registrador inicial (byte low)	Dado 1 (high)
Número de registradores (byte high)	Dado 1 (low)
Número de registradores (byte low)	Dado 2 (high)
CRC-	Dado 2 (low)
CRC+	etc...
	CRC-
	CRC+

☑ Exemplo: leitura dos valores de velocidade (P002) e corrente do motor (P003) do CFW-09 no endereço 1:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	01h	Endereço do escravo	01h
Função	03h	Função	03h
Registrador inicial (high)	00h	Byte Count	04h
Registrador inicial (low)	02h	P002 (high)	03h
No. De registradores (high)	00h	P002 (low)	84h
No. De registradores (low)	02h	P003 (high)	00h
CRC-	65h	P003 (low)	35h
CRC+	CBh	CRC-	7Ah
		CRC+	49h

Observação importante sobre marcadores tipo FLOAT:

Como um marcador float tem 4 bytes, o mestre deverá requisitar dois registros para ler um float, por exemplo:

Para ler o MF9000, o endereço é 9000 e a quantidade deve ser 2, ou seja, será retornado 4 bytes (2 words), que representam o marcador float no formato IEEE 754 (IEEE Standard Floating Point Format).

Se for pedida uma quantidade ímpar de registros, será retornado erro 2.

12.1.3.3 Função 04 - Read Input Register

Lê o conteúdo das entradas analógicas do drive. Esta função possui a seguinte estrutura para os telegramas de leitura e resposta (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do registrador inicial (byte high)	Campo Byte Count
Endereço do registrador inicial (byte low)	Dado 1 (high)
Número de registradores (byte high)	Dado 1 (low)
Número de registradores (byte low)	Dado 2 (high)
CRC-	Dado 2 (low)
CRC+	Etc...
	CRC-
	CRC+

Exemplo: leitura das entradas analógicas 101 e 102 do CFW-09 no endereço 1:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	01h	Endereço do escravo	01h
Função	04h	Função	04h
Registrador inicial (high)	00h	Byte Count	04h
Registrador inicial (low)	65h	AI101 (high)	1Fh
No. De registradores (high)	00h	AI101 (low)	A0h
No. De registradores (low)	02h	AI102 (high)	0Dh
CRC-	61h	AI102 (low)	20h
CRC+	D4h	CRC-	F9h
		CRC+	3^{A}h

Cada registrador sempre é formado por dois bytes (high e low). Para o exemplo, temos que AI101 = 1FA0h, que em decimal é igual a 8096, e AI102 = 0D20h = 3360. Como as entradas analógicas variam entre 0 e 32767, essa leitura representa respectivamente, 24,7% e 10,25% do valor de fundo de escala das entradas.

Observação: não esquecer que as entradas analógicas do drive (AI1 e AI2) são representadas na PLC como AI101 e AI102, respectivamente.

12.1.3.4 Função 05 - Write Single Coil

Esta função é utilizada para escrever um valor em um marcador de bit, marcador de sistema ou saída digital. O valor do bit é representado utilizando dois bytes, onde o valor FF00h representa o bit igual a 1, e o valor 0000h representa o bit igual a 0 (zero). Possui a seguinte estrutura (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do bit (byte high)	Endereço do bit (byte high)
Endereço do bit (byte low)	Endereço do bit (byte low)
Valor para o bit (byte high)	Valor para o bit (byte high)
Valor para o bit (byte low)	Valor para o bit (byte low)
CRC-	CRC-
CRC+	CRC+

Exemplo: acionar a saída digital 2 da PLC no endereço 1:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	01h	Endereço do escravo	01h
Função	05h	Função	05h
No. Do bit (high)	00h	No. Do bit (high)	00h
No. Do bit (low)	02h	No. Do bit (low)	02h
Valor para o bit (high)	FFh	Valor para o bit (high)	FFh
Valor para o bit (low)	00h	Valor para o bit (low)	00h
CRC-	2Dh	CRC-	2Dh
CRC+	Fah	CRC+	Fah

Para esta função a resposta do escravo é uma cópia idêntica da solicitação feita pelo mestre.

12.1.3.5 Função 06 - Write Single Register

Esta função é utilizada para escrever um valor em um marcador word ou parâmetro da PLC ou do drive. Não pode ser utilizada para escrita em marcador float. Possui a seguinte estrutura (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do parâmetro (byte high)	Endereço do parâmetro (byte high)
Endereço do parâmetro (byte low)	Endereço do parâmetro (byte low)
Valor para o parâmetro (byte high)	Valor para o parâmetro (byte high)
Valor para o parâmetro (byte low)	Valor para o parâmetro (byte low)
CRC-	CRC-
CRC+	CRC+

Exemplo: escrita da referência de velocidade igual a 900 rpm, em um parâmetro do usuário (P800) no endereço 1.

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	01h	Endereço do escravo	01h
Função	06h	Função	06h
Parâmetro (high)	03h	Parâmetro (high)	03h
Parâmetro (low)	20h	Parâmetro (low)	20h
Valor (high)	03h	Valor (high)	03h
Valor (low)	84h	Valor (low)	84h
CRC-	88h	CRC-	88h
CRC+	D7h	CRC+	D7h

Para esta função a resposta do escravo é uma cópia idêntica da solicitação feita pelo mestre. Os marcadores word ou parâmetros são endereçados diretamente pelo seu número, no exemplo acima P800 = 0320h.

12.1.3.6 Função 15 - Write Multiple Coils

Esta função permite escrever valores para um grupo de marcadores bit ou saídas digitais da PLC ou do drive, que devem estar em seqüência numérica. Também pode ser usada para escrever em um único bit (os valores são sempre hexadecimal, e cada campo representa um byte).

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do bit inicial (byte high)	Endereço do bit inicial (byte high)
Endereço do bit inicial (byte low)	Endereço do bit inicial (byte low)
Número de bits (byte high)	Número de bits (byte high)
Número de bits (byte low)	Número de bits (byte low)
Campo Byte Count (no. de bytes de dados)	CRC-
Byte 1	CRC+
Byte 2	
Byte 3	
etc...	
CRC-	
CRC+	

O valor de cada bit que está sendo escrito é colocado em uma posição dos bytes de dados enviados pelo mestre. O primeiro byte, nos bits de 0 a 7, recebe os 8 primeiros bits a partir do endereço inicial indicado pelo mestre. Os demais bytes (se o número de bits escritos for maior que 8), continuam em seqüência. Caso o número de bits escritos não seja múltiplo de 8, os bits restantes do último byte devem ser preenchidos com 0 (zero).
Exemplo: ligar as saídas digitais 4 e 5 da PLC, no endereço 1:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	01h	Endereço do escravo	01h
Função	0Fh	Função	0Fh
Bit inicial (byte high)	00h	Bit inicial (byte high)	00h
Bit inicial (byte low)	04h	Bit inicial (byte low)	04h
No. de bits (byte high)	00h	No. de bits (byte high)	00h
No. de bits (byte low)	02h	No. de bits (byte low)	02h
Byte Count	01h	CRC-	95h
Valor para os bits	03h	CRC+	CBh
CRC-	6Fh		
CRC+	56h		

Como estão sendo escritos apenas dois bits, o mestre precisou de apenas 1 byte para transmitir os dados. Os valores transmitidos estão nos dois bits menos significativos do byte que contém o valor para os bits. Os demais bits deste byte foram deixados com o valor 0 (zero).

12.1.3.7 Função 16 - Write Multiple Registers

Esta função permite escrever valores para um grupo de marcadores word, marcadores float, parâmetros da PLC ou do drive, que devem estar em seqüência numérica. Também pode ser usado para escrever um único parâmetro (os valores são sempre hexadecimal, e cada campo representa um byte).

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do parâmetro inicial (byte high)	Endereço do parâmetro inicial (byte high)
Endereço do parâmetro inicial (byte low)	Endereço do parâmetro inicial (byte low)
Número de parâmetros (byte high)	Número de parâmetros (byte high)
Número de parâmetros (byte low)	Número de parâmetros (byte low)
Campo Byte Count (nº de bytes de dados)	CRC-
Dado 1 (high)	CRC+
Dado 1 (low)	
Dado 2 (high)	
Dado 2 (low)	
etc...	
CRC-	
CRC+	

Exemplo: escrita do tempo de aceleração (P100) = 1,0 s e tempo de desaceleração (P101) = 2,0 s, de um CFW-09 no endereço 20:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	*Valor*	*Campo*	*Valor*
Endereço do escravo	14h	Endereço do escravo	14h
Função	10h	Função	10h
Registrador inicial (high)	00h	Registrador inicial (high)	00h
Registrador inicial (low)	64h	Registrador inicial (low)	64h
No. De registradores (high)	00h	No. De registradores (high)	00h
No. De registradores (low)	02h	No. De registradores (low)	02h
Byte Count	04h	CRC-	02h
P100 (high)	00h	CRC+	D2h
P100 (low)	0Ah		
P101 (high)	00h		
P101 (low)	14h		
CRC-	91h		
CRC+	75h		

Observação importante sobre marcadores tipo FLOAT:

Como um marcador float tem 4 bytes, o mestre deverá enviar dois registros para escrever em um float, por exemplo:

Para escrever no MF9000, o endereço é 9000 e a quantidade deve ser 2, ou seja, serão enviados 4 bytes (2 words), que representam o marcador float no formato IEEE 754 (IEEE Standard Floating Point Format).

Se for enviada uma quantidade ímpar de registros, será retornado erro 2.

Como ambos os parâmetro possuem resolução de uma casa decimal, para escrita de 1,0 e 2,0 segundos, devem ser transmitidos respectivamente os valores 10 (000Ah) e 20 (0014h).

12.1.3.8 Função 43 - Read Device Identification

Função auxiliar, que permite a leitura do fabricante, modelo e versão de firmware do produto. Possui a seguinte estrutura:

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
MEI Type	MEI Type
Código de leitura	Conformity Level
Número do Objeto	More Follows
CRC-	Próximo Objeto
CRC+	Número de objetos
	Código do Objeto*
	Tamanho do Objeto*
	Valor do Objeto*
	CRC-
	CRC+

* Campos são repetidos de acordo com o número de objetos.Esta função permite a leitura de três categorias de informações: Básicas, Regular e Estendida, e cada categoria é formada por um grupo de objetos. Cada objeto é formado por um seqüência de caracteres ASCII. Para a PLC, apenas informações básicas estão disponíveis, formadas por três objetos:þ
Objeto 00 - VendorName: Sempre 'WEG'.þ Objeto 01 - ProductCode: Formado pelo código do produto (PLC1.01) onde 01 indica a versão de hardware.

Objeto 02 - MajorMinorRevision: indica a versão de firmware da PLC, no formato 'VX.XX'.O código de leitura indica quais as categorias de informações estão sendo lidas, e se os objetos estão sendo acessados em seqüência ou individualmente. No caso, a PLC suporta os códigos 01 (informações básicas em seqüência), e 04 (acesso individual aos objetos).

Exemplo: leitura das informações básicas em seqüência, a partir do objeto 00, de uma PLC no endereço 1:

12.1.4 Erro de Comunicação

Os erros podem ocorrer na transmissão dos telegramas na rede, ou então no conteúdo dos telegramas recebido. De acordo com o tipo de erro, a PLC poderá ou não enviar resposta para o mestre:
Quando o mestre envia uma mensagem para a placa configurada em um determinado endereço da rede, a mesma não irá responder ao mestre caso ocorra:

- ☑ Erro no CRC.
- ☑ Time out entre os bytes transmitidos (3,5 vezes o tempo de transmissão de uma palavra de 11 bits).

No caso de uma recepção com sucesso, durante o tratamento do telegrama, a PLC pode detectar problemas e enviar uma mensagem de erro, indicando o tipo de problema encontrado:

- ☑ Função inválida (código do erro = 1): a função solicitada não está implementada para a PLC.
- ☑ Endereço de dado inválido (código do erro = 2): o endereço do dado (parâmetro ou E/S digital) não existe.
- ☑ Valor de dado inválido (código do erro = 3): ocorre nas seguintes situações:
  - ☑ Valor está fora da faixa permitida.
  - ☑ Escrita em dado que não pode ser alterado (registrador somente leitura, registrador que não permite alteração com o conversor habilitado ou bits do estado lógico).
  - ☑ Escrita em função do comando lógico que não está habilitada via serial.

12.1.4.1 Mensagens de Erro

Quando ocorre algum erro no conteúdo da mensagem (não na transmissão de dados), o escravo deve retornar uma mensagem que indica o tipo de erro ocorrido. Os erros que podem ocorrer no tratamento de mensagens para a PLC são os erros de função inválida (código 01), endereço de dado inválido (código 02) e valor de dado inválido (código 03).
As mensagens de erro enviadas pelo escravo possuem a seguinte estrutura:

Resposta (Escravo)
Endereço do escravo
Código da função(com o bit mais significativo em 1)
Código do erro
CRC-
CRC+